SIS8300-KU
10 channel
125 MSPS 16-bit
MTCA.4 Digitizer

User Manual

SIS GmbH
Harksheider Str. 102A
22399 Hamburg
Germany

Phone: ++49 (0) 40 60 87 305 0
Fax: ++49 (0) 40 60 87 305 20

email: info@struck.de
http://www.struck.de

Version: SIS8300KU-M-x002-1-V101.doc as of 09.03.2017

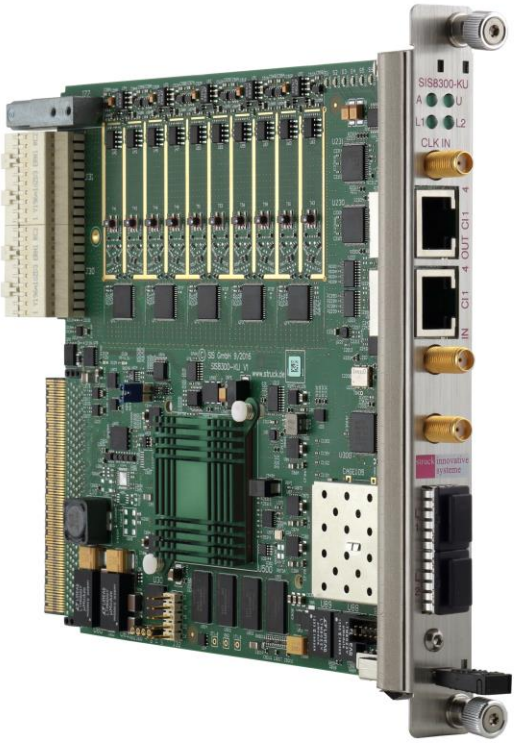
Revision Table:

Revision	Date	Modification
0.01	24.10.2016	Based on SIS8300L2-M-x00D-1-V110 (Firmware: V100D)
0.02	25.10.2016	Chapter "Functionality" updated Chapter "Block diagram" updated Chapter "Platform Management" MMC scheme updated Chapter "MGT clock" scheme updated Chapter "Frontpanel" sketch updated Chapter 5.1 changed from Harlink to "RJ45 In-/Outputs" and updated Chapter "Board Layout" print and used connector table updated Chapter "Firmware Options register" updated Chapter "RJ45 connector... register" updated Chapter "SIS8900 RTM LVDS ... register" updated Chapter "Ordering Options" updated Chapter "Zone 3 connector schematic" updated
0.03	09.11.2016	Chapter "Functionality" updated
0.04	12.12.2016	Register description update
0.05		Block diagram update
0.06	15.12.2016	Chapter "Overall Clock distribution" updated Chapter "Interlock" updated Chapter "SIS8900 RTM LVDS ... register" updated
1.00	13.01.2017	First official release Based on firmware sis8300KU_x001
1.01	09.03.2017	Release related to firmware sis8300KU_x002 (IP sis8300KU_x802) Added feature in firmware: - FPGA Boot SPI Flash interface to update the FPGA Boot Flash via PCIe

- Table of contents

1 Introduction

The SIS8300-KU is a 10 channel 125 MS/s digitizer with 16-bit resolution according to the MTCA.4 standard.



SIS8300-KU

Note: While the SIS8300-KU is Kintex Ultrascale based you will find many Virtex 5/6 references in the firm- and software for historical reasons (i.e. remnants from the SIS8300_V2 and SIS8300L2 designs).

As we are aware, that no manual is perfect, we appreciate your feedback and will incorporate proposed changes and corrections as quickly as possible. The most recent version of this manual can be obtained from our Dokuwiki at www.struck.de/dokuwiki. Please direct account requests to info@struck.de.

1.1 Related documents

A list of available firmware designs can be retrieved from http://www.struck.de/SIS8300firm.html

2 Design

The central building block of the SIS8300-KU card is a Xilinx Kintex Ultrascale FPGA. It holds the 4 lane PCI Express interface and is in control of all active components.

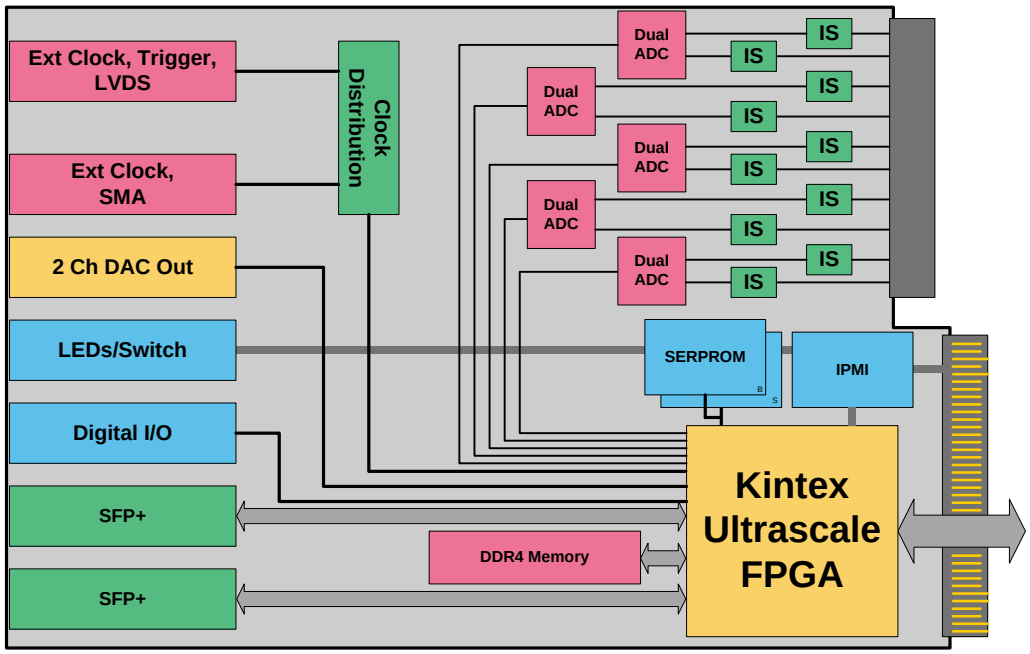
2.1 Functionality

The key properties of the SIS8300-KU card are listed below.

- AMC .4 µTCA for Physics Board
- 4 Lane PCI Express Gen3 Interface
- Dual zSFP+ Card Cage for optional Multi Gigabit Link
- Xilinx Kintex Ultrascale FPGA
- DDR4 Memory Interface
- 4 x 4Gbit default DDR4 memory size
- ATxmega128A1U Microcontroller IPMI
- External Clock and Trigger Inputs
- Front panel digital I/O (4in/4 out) on RJ45 Connectors
- µRTM ADC Analog Inputs, I²C-Bus, DAC Analog Outputs
- 10 ADC Channels 125MS/s, 16-Bit
- 2 DAC Channels 250MS/s, 16-Bit
- Clock distribution with phase shifting
- 4 M-LVDS µTCA Ports
- 2 µTCA Clocks
- White Rabbit Option (depends on assembly)

2.2 Block Diagram

A simplified block diagram of the SIS8300-KU is shown below.

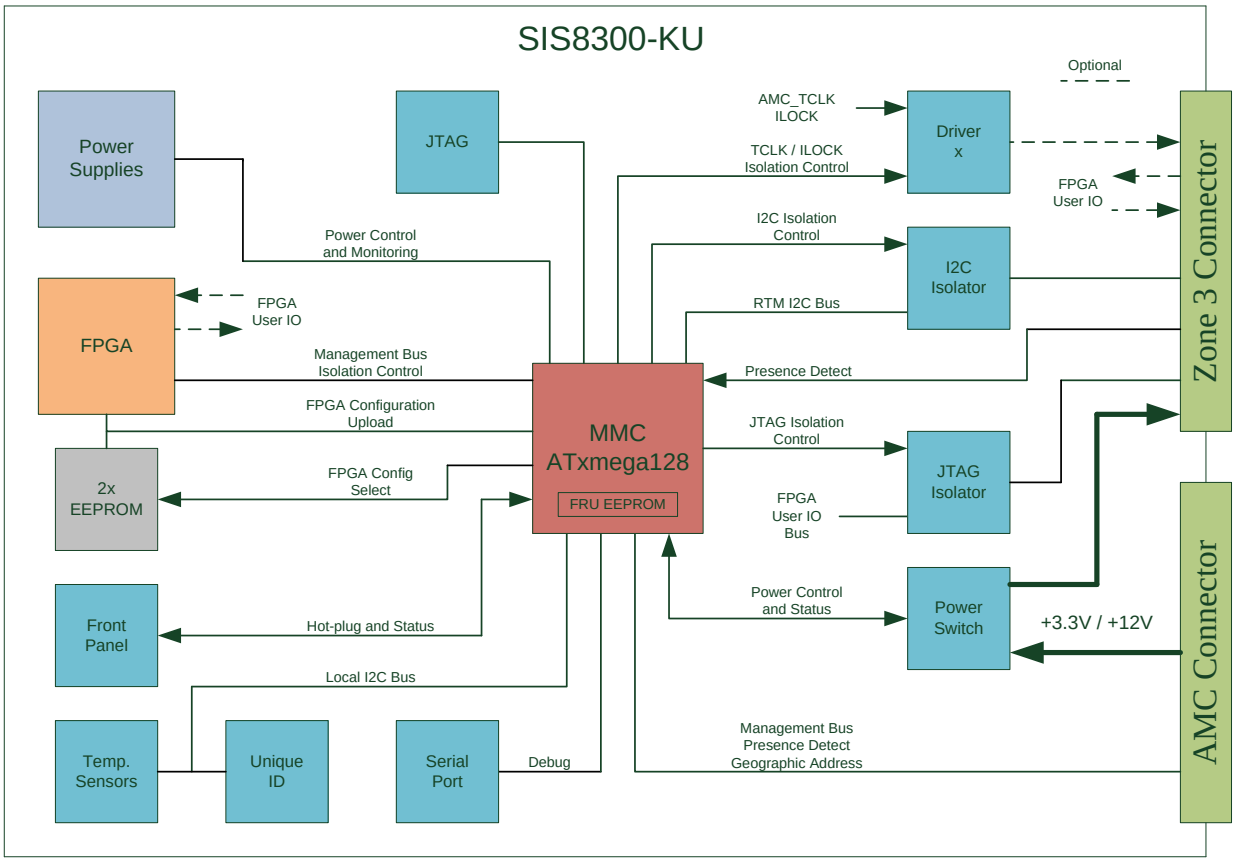


2.3 Platform Management

The Module Management Controller (MMC) functionality of the SIS8300-KU is implemented in an Atmel ATxmega128A1U-CU microcontroller. The management code is based on the DESY MMC V1.00 management software solution(*). It can be upgraded in two different ways (please refer to section 2.3.3).

(*) The DESY Module Management Controller (MMC) software is licensed to Struck Innovative Systeme GmbH under DESY LV92. Please, refer to the readme file in the appendix of this manual also.

The MMC scheme of the SIS8300-KU is illustrated below.



2.3.1 E-Keying

An Electronic Keying mechanism is used to provide correct connection and setup of point-to-point fabrics and clocks for AMC modules in example. E-Keying information of the SIS8300-KU is stored in the MMC as FRU information. It will be communicated to the MCH at power up. E-Keying entries in the FRU information of the SIS8300-KU match the FPGA design at shipment version and are fixed. If you change the FPGA design and/or make a custom FPGA design it can be necessary to change part of the FRU information also.
The use of incorrect FRU information can result in unexpected behaviour.

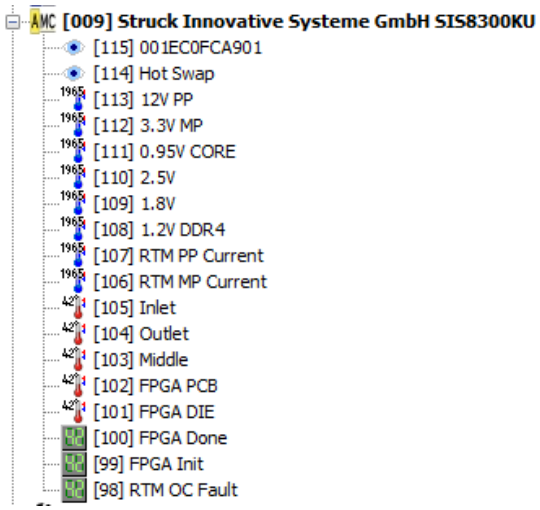
Please contact us in case you need detailed information on SIS8300-KU E-Keying.

More information on this issue is planned for future additions to the documentation.

2.3.2 MMC sensors

Different sensors are connected to the MMC. They are giving information about temperatures, voltages, currents and states. An additional memory with unique ID is available for board identification purposes.

An overview of all existing SIS8300-KU MMC sensors –as seen with NATView- is illustrated below.



Note: On SIS8300-KU further temperature sensors exist, please see section 2.6.

2.3.3 MMC firmware upgrade

The MMC firmware can be upgraded in field over connector J32 or can be uploaded via IPMI.

To upgrade in field you need an AVR programming tool, i.e. AVR JTAGICE mkII. After connecting to connector J32 (please refer to section 3.3) the MMC can be upgraded using Atmel-Studio software.

The more comfortable means for an upgrade is the upload of the MMC code over IPMI. This can be done with the LINUX software ipmitool.

Syntax:

```
ipmitool -I lan -H <IP-MCH> -P "" -B 0 -b 7 -T 0x82 -t <Slot> ↵
  hpm upgrade SIS8300KU_VersionID.hpm
```

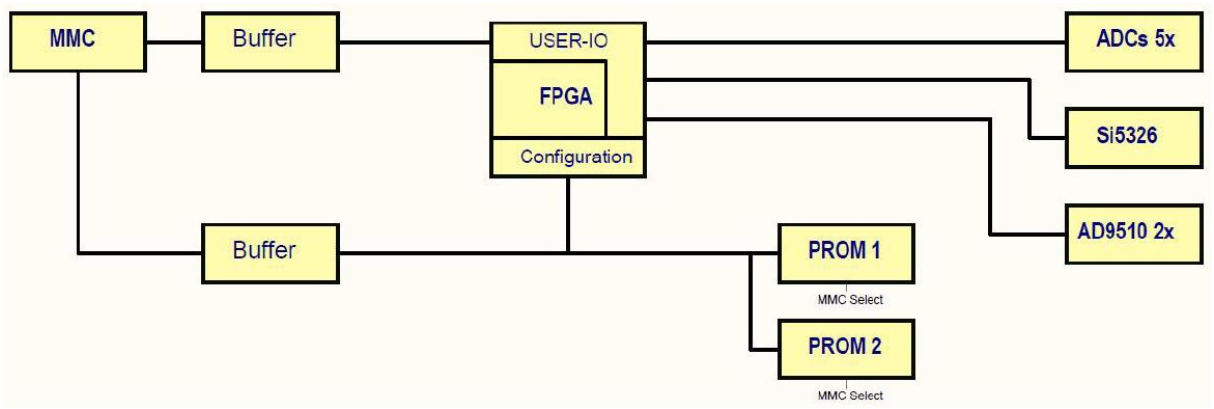
`Slot`: 0x72 (1st slot), 0x74 (2nd slot), 0x76 (3rd slot), ...

for example:
ipmitool -H 192.168.115.62 -P "" -B 0 -b 7 -T 0x82 -t 0x74 hpm upgrade SIS8300KU_VersionID.hpm
ipmitool -H 192.168.115.62 -P "" -B 0 -b 7 -T 0x82 -t 0x74 mc info

To verify, that the MMC was upgraded successfully, mc info can be read to check the MMC code version.

2.4 SPI Connectivity

The SPI bus connectivity scheme of the SIS8300-KU is illustrated below.



The SIS8300-KU has two SPI EEPROMs, that can hold FPGA configuration data. So it is possible to choose from two firmware designs for the configuration of the FPGA. The selection of the SPI EEPROM is under control of the MMC and can be changed per IPMI command.

The ipmitool command line to set one of the Flashes as boot file source looks like:

```
ipmitool -I lan -H <IP-MCH> -P "" -B 0 -b 7 -T 0x82 -t <Slot> ↵
  raw 0x30 0x01 <Flash>
```

`Flash`: 0x00 (Basic-FLASH), 0x01 (Second-FLASH)
`Slot`: 0x72 (1st slot), 0x74 (2nd slot), 0x76 (3rd slot), ...

Verification:

```
ipmitool -I lan -H <IP-MCH> -P "" -B 0 -b 7 -T 0x82 -t <Slot> ↵
  raw 0x30 0x00
```

Result:

```
00    : Basic-FLASH selected
01    : Second-FLASH selected
```

IPMI raw command to reset the FPGA:

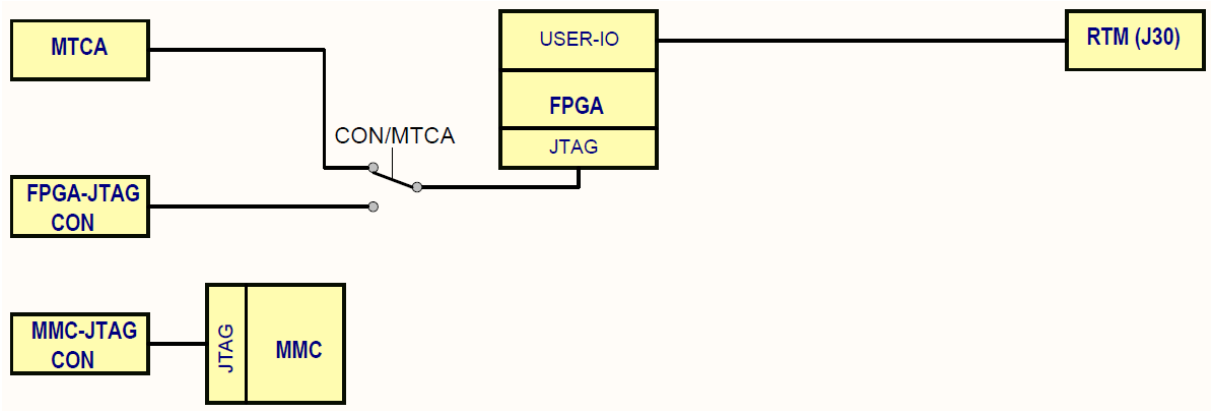
```
ipmitool -I lan -H <IP-MCH> -P "" -B 0 -b 7 -T 0x82 -t <Slot> ↵
  raw 0x30 0xFF
```

The user has to take special care to avoid concurrent access to the SPI EEPROM from different sides (via PCIe and IPMI).

The SPI EEPROMs can be flashed via a JTAG-Programmer, PCIe or IPMI with appropriate tools. Please refer to section 9 for more information about flashing the SPI EEPROM.

2.5 JTAG Connectivity

The JTAG connectivity scheme of the SIS8300-KU is illustrated below.



JTAG connection to µRTM can be realized in FPGA firmware but is a optional feature for future.
Switching connection of FPGA hardware JTAG port between MTCA (AMC Backplane) and on board JTAG connector (illustrated switch CON/MTCA) can be made via IPMI command.

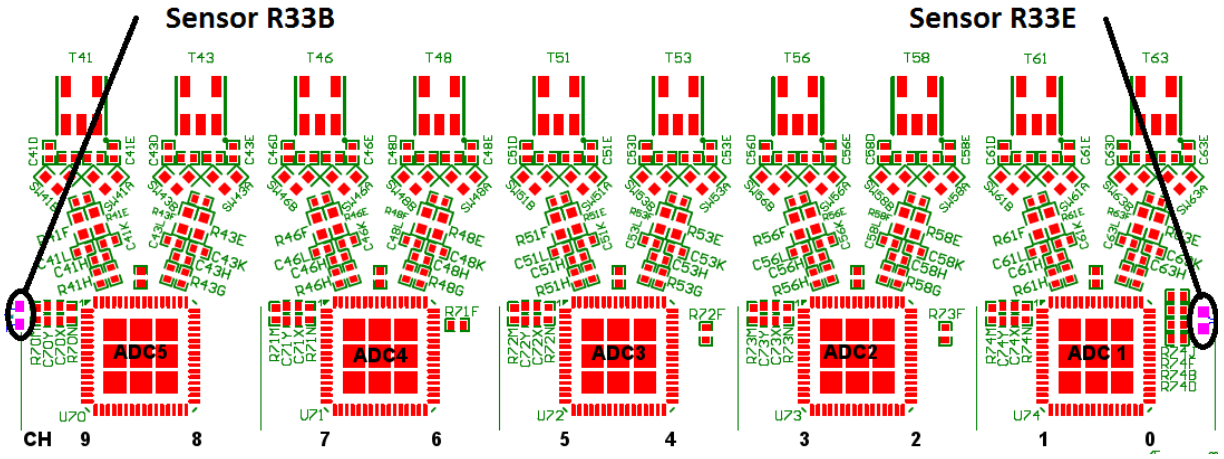
IPMI command ... (not implemented yet, please request us for more information)

Upon power up the switch CON/MTCA connects the JTAG connector to the JTAG port of the FPGA hardware.

2.6 ADC Temperature Sensors

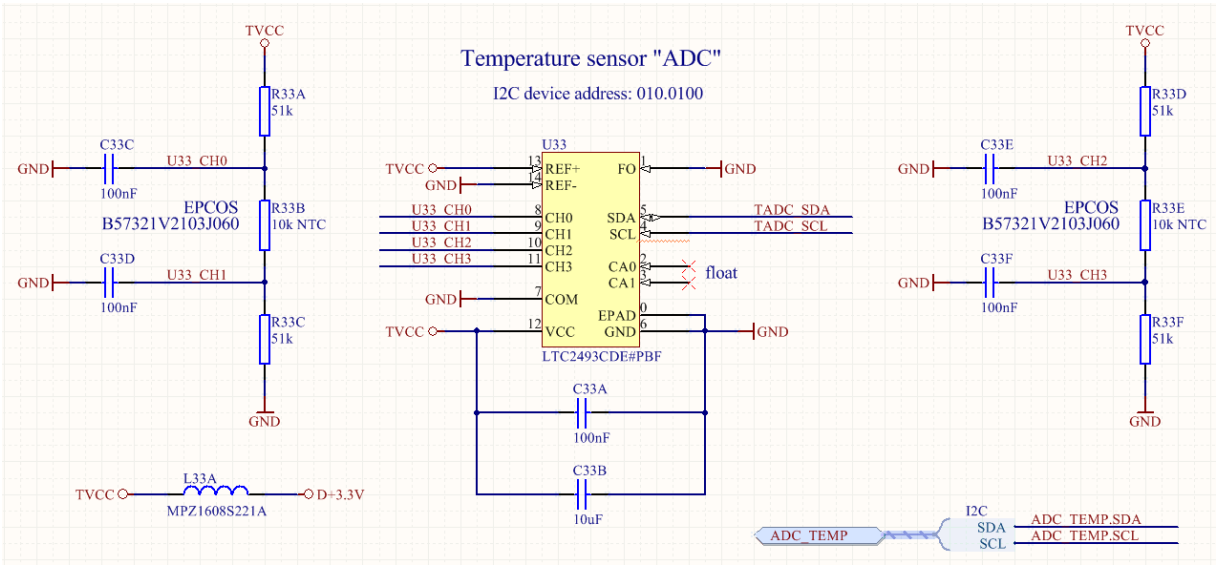
Two thermistor based sensors are used to measure the temperature at the beginning and the end of the row of the 5 digitizer chips.

The location of both sensors on the bottom side as seen from the top is shown below.



The sensor signals are measured by the 24-bit ΣΔ-ADC LTC2493, which is connected to the Kintex Ultrascale FPGA via an I²C interface.

The schematic of the ADC temperature sensor circuitry is illustrated below.



Note: please refer to section 7.5.5 for a description of the I²C interface register

Please refer to device datasheet for additional information on the LTC2493 chip.
A software example can be found on the Struck product DVD under:
sisdvd_xxxxxx\sis8xxx and DWC\sis8300L\software\tests\adc_temp

2.7 DAC

A part of the SIS8300-KU versions comes with the 250 MSPS MAX5878 dual DAC chip. The dual DAC can be used to control the Vectormodulator on the DWC8VM1 Downconverter/Vectormodulator RTM over the Zone 3 connector for example. The two DAC channels can be routed to two front panel SMA outputs also (stuffing option). The table in section 10.2 (ordering options) lists a number of DAC configurations of the SIS8300-KU.

Modules configured as DAC to FP outputs route the two DAC outputs over signal conditioning stages to the two front panel SMA connectors. The default range of the DAC outputs over the SMA connectors is -1V,…,+1V into a 50 Ω load.

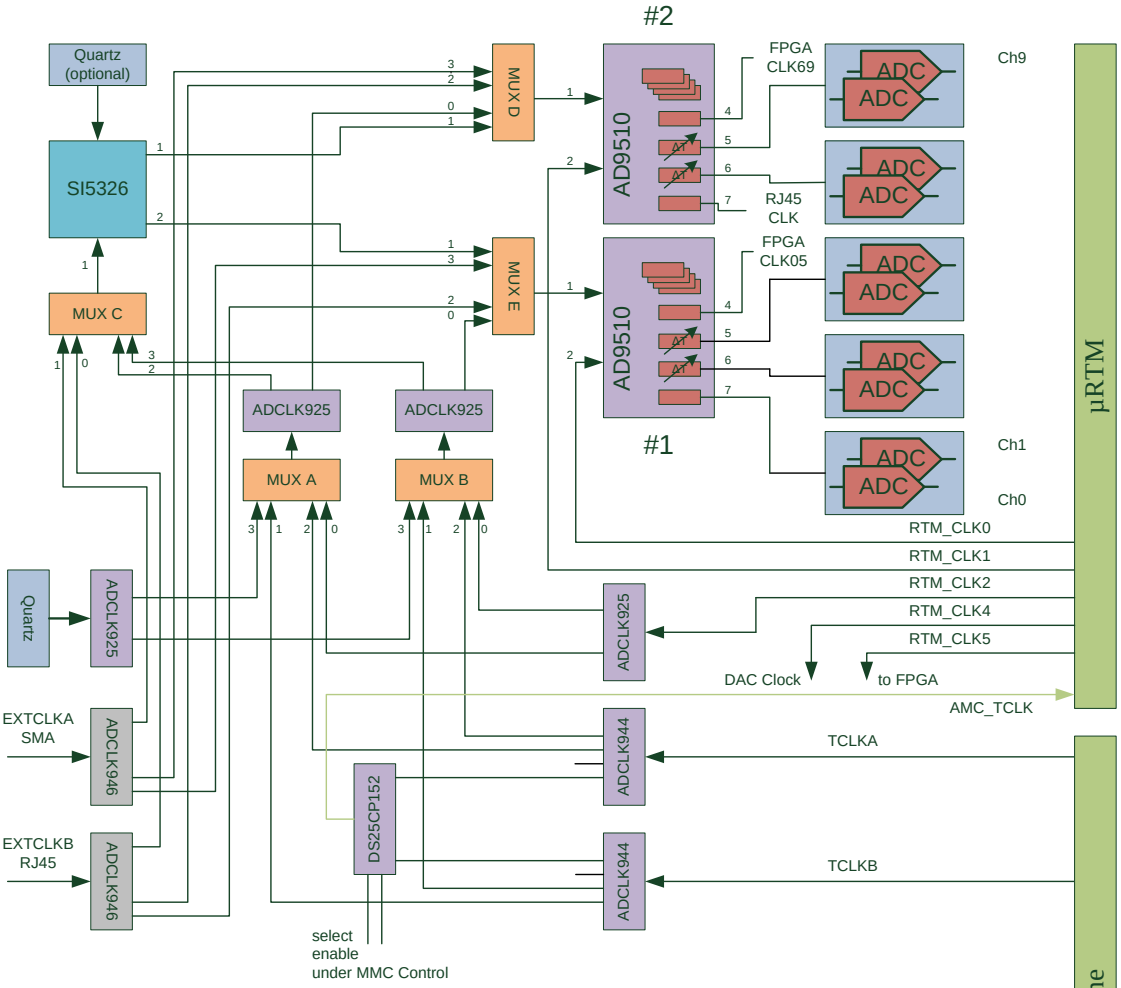
On modules configured with DAC to Z3, the outputs of the DAC are directly connected to the Zone 3 connector without any additional components. Since each DAC channel outputs two complementary currents, appropriate signal conditioning has to be done on the µRTM.

Refer to the datasheet of the MAX5878 for further information.

2.8 Clock Distribution

2.8.1 Overall Clock Distribution

The clock distribution scheme of the SIS8300-KU is illustrated below.



Inputs:

MUX D	→	AD9510-#2_CLK1
RTM_CLK1	→	AD9510-#2_CLK2
MUX E	→	AD9510-#1_CLK1
RTM_CLK0	→	AD9510-#1_CLK2

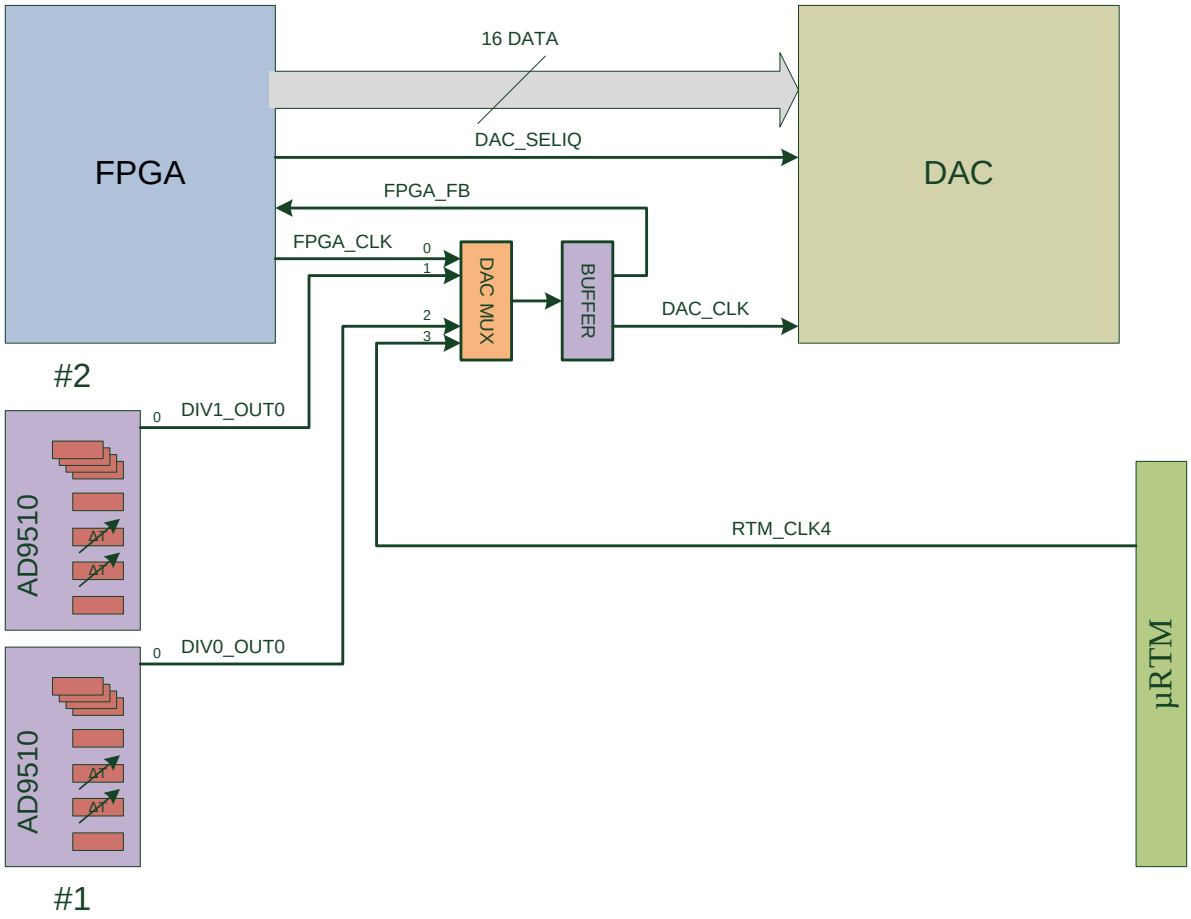
Outputs:

AD9510-#2_OUT7 →	FP_RJ45_E
AD9510-#2_OUT6 →	ADC4_CLK
AD9510-#2_OUT5 →	ADC5_CLK
AD9510-#2_OUT4 →	FPGA_CLK69
AD9510-#2_OUT3 →	N/A
AD9510-#2_OUT2 →	N/A
AD9510-#2_OUT1 →	FPGA_MGT_CLK
AD9510-#2_OUT0 →	DIV1_OUT0 (DAC_MUX)
AD9510-#1_OUT7 →	ADC1_CLK
AD9510-#1_OUT6 →	ADC2_CLK
AD9510-#1_OUT5 →	ADC3_CLK
AD9510-#1_OUT4 →	FPGA_CLK05
AD9510-#1_OUT3 →	N/A
AD9510-#1_OUT2 →	N/A
AD9510-#1_OUT1 →	N/A
AD9510-#1_OUT0 →	DIV0_OUT0 (DAC_MUX)

Depending from ordering options a 114.285 MHz clock will be populated. This external reference is required for the SI5326 to perform jitter attenuation. Please refer to the SI5326 datasheet and section 10.2.

2.8.2 DAC Clock

The DAC clock scheme of the SIS8300-KU is illustrated below.



2.8.3 µRTM Clock Overview

The µRTM clock overview table is shown below.

Clock	Usage
RTM_CLK0	Ultra low jitter clock 1 ADC group 1
RTM_CLK1	Ultra low jitter clock 2 ADC group 2
RTM_CLK2	Clock switch yard
RTM_CLK3	Not used
RTM_CLK4	DAC Clock
RTM_CLK5	Connected to FPGA via clock buffer

2.8.4 TCLK Clock Overview

An overview on the TCLK clocks is shown in the table below.

Clock	Usage
TCLKA	MUX switch yard and Zone 3 AMC_TCLK
TCLKB	MUX switch yard and Zone 3 AMC_TCLK
TCLKC	Not used, can be 100 Ω terminated if needed
TCLKD	Not used, can be 100 Ω terminated if needed

2.8.5 AMC_TCLK on Zone 3

The AMC_TCLK signal can be derived from TCLKA or TCLKB. Selection and output control of AMC_TCLK switch is under control of MMC. Currently the state of both signals can be read from SIS8900 RTM LVDS Test Input/Output Control register, please refer to section 7.5.32.

The TCLKA or TCLKB selection and switch output control can be made per IPMI command:

```
<IPMI> raw 0x30 0x03 0xEC
```
(E=1/0: TCLK active/deactivated, C=0/1: select TCLKA/TCLKB)

The status can be read back with the command:

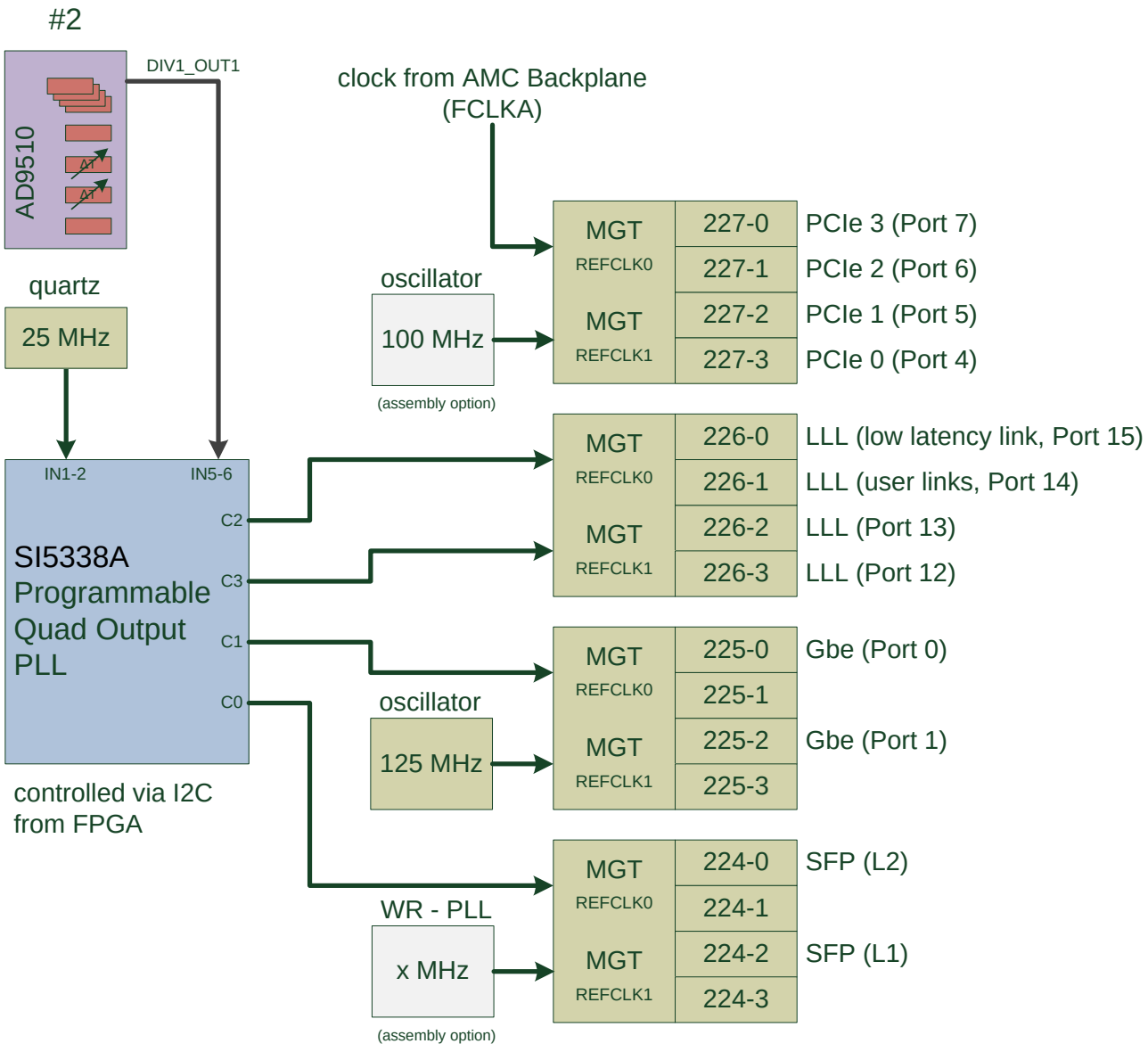
```
<IPMI> raw 0x30 0x02
```

After power up TCLKA is selected per default.

AMC_TCLK signal is available for SIS8300-KU with Zone 3 class compatibility A1.1CO and A1.0C only. Please refer to section 10.4.2.

2.8.6 MGT Clock

The MGT (Multi Gigabit Transceiver) clock scheme of the SIS8300-KU is illustrated below.

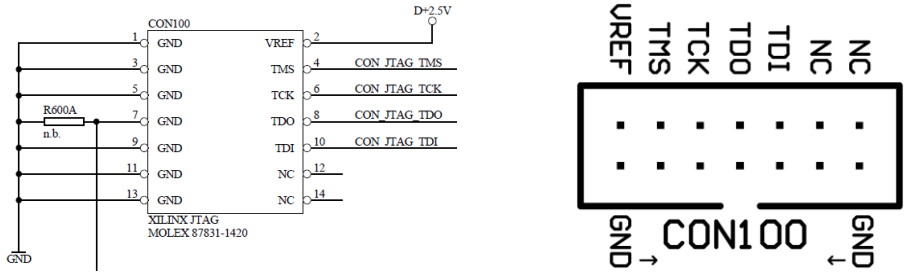


3 Jumper/Connector Pin Assignments

The following subsections describe jumper and connector pin assignments.

3.1 CON100 JTAG

The SIS8300-KU's on board logic can load its firmware from two SPI EEPROMs (Basic FLASH and Second FLASH), via the JTAG port on connector CON100, PCI Express, MTCA or via the MMC.
Hardware like the XILINX HW-USB-JTAG in connection with the appropriate software will be required for in field JTAG firmware upgrades.
CON100 is a 2mm (i.e. metric) 14 pin header that allows you to reprogram the firmware of the SIS8300-KU board with a JTAG programmer. The pin out is shown in the schematic below. It is compatible with the cable that comes with the XILINX HW-USB-II-G-JTAG platform cable. CON100 can be found at the right bottom side of the board.



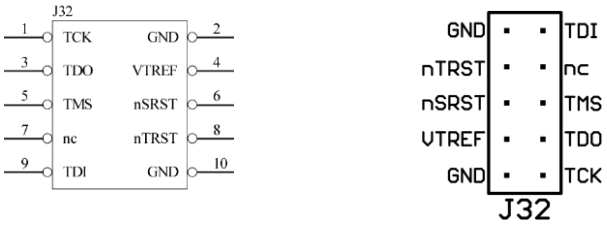
Note: The board has to be powered for reprogramming over JTAG

3.2 J604 Watchdog Reset

J604 can be found next to the left upper edge of U500 (largest chip on the card with green heat sink). With J604 closed the boards watchdog reset is connected to the reset logic. J604 should be opened for JTAG firmware programming.

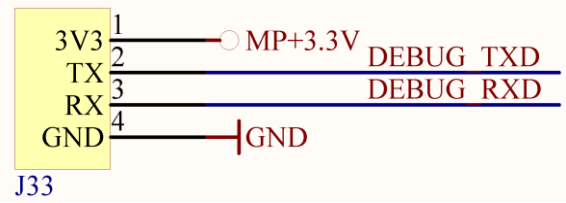
3.3 J32 AVR JTAG

This 10-pin header is used to connect to the JTAG of the Atmel ATxmega128 microcontroller providing the IPMI/MCH functionality of the SIS8300-KU. J32 can be found at the left bottom side of the board. The pin out is shown in the schematic below.



3.4 J33 ATxmega128 Debug

This 4 pin socket strip grants access to the debug port (PE2 and PE3) of the ATxmega128 microcontroller. It was designed in to facilitate Struck in house microcontroller software development.



Note: Due to a lack of coding there is no protection against a wrong polarity connection. This connector is reserved for Struck in house purposes only.

4 LEDs

4.1 Module Management LEDs

Three right angle front panel LEDs (blue, red, green, not labelled) show the hot-swap status and general module information about failures and out of service status.
The front panel Hot Swap handle has to be used for module insertion and extraction.
To insert the card, the Hot Swap handle must be pulled. The Hot Swap handle can pushed in once the module is properly seated in the carrier backplane connector to initiate the MCH to bring the module into operational state.

Typical module insertion process (module handle pushed in)

blue	red	green	Status
On	Off	Off	Module management controller start, activation process begins
Blink	Off	Off	Module activation process in progress, wait for payload power
Blink	On	Off	Payload power present, MMC waits for onboard Power Good
Off	Off	On	Power Good, FPGA loaded (done signal), module operational

Typical module extraction process (module handle pulled out)

blue	red	green	Status
Blink	Off	On	Module still operational, deactivation process begins, wait ...
Blink	On	Off	Payload power off, Module deactivation process in progress
On	Off	Off	Module deactivation finished and safe for extraction

Note: do not remove module before Hot Swap handle is pulled and the blue LED is solid on

4.2 Front Panel LEDs

The SIS8300-KU has additional 4 circular green front panel LEDs and 4 right angled green LEDs next to dual SFP card cage. The function of the LEDs are shown in the table below.

LED name	Function in base design
A	PCI Express Access
U	User LED
L1	PCIe Link up
L2	ADC Sampling active
R (L1)	SFP Link 1 receiver loss of signal
T (L1)	SFP Link 1 transmitter fault
R (L2)	SFP Link 2 receiver loss of signal
T (L2)	SFP Link 2 transmitter fault

Note: if SIS8300-KU is in operational state and no SFP modules are present in SFP card cage associated LEDs (R and T) are lit permanently

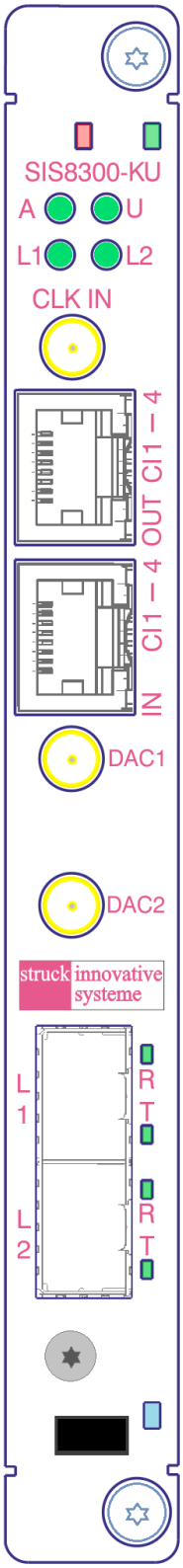
4.3 SMD LEDs

A number of surface mount red LEDs are on the SIS8300-KU to visualize part of the board status.

LED designator	LED comment	Function
D20A	S1	Firmware dependent (Optical Link 1 up)
D20B	S2	Firmware dependent (Optical Link 2 up)
D20C	S3	Firmware dependent
D20D	S4	Firmware dependent
D20E	S5	Firmware dependent
D20F	S6	Firmware dependent
D20G	S7	Firmware dependent
D20H	S8	Firmware dependent
D21D	READY	FPGA ready
D10A	+12V	Payload power present
D10B	MP+3.3V	Management power present

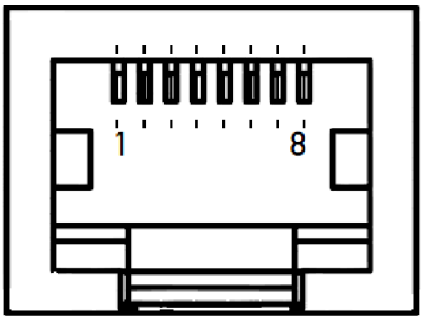
5 Front panel

A sketch of the SIS8300-KU front panel view (with front panel I/O option) is shown below.



5.1 RJ45 LVDS In-/Outputs

Two identical RJ45 connectors are present for clock and digital I/O signals.
A drawing with the pin count and pin orientation is shown below.



Front view

The RJ45 Input connector has inputs only and RJ45 Output connector has outputs respectively. Both connectors have 8 pins for 4 differential signals.

In table below clock signal is marked with C and data signals are marked with D0-3. Each differential signal consists of the complementary signals marked with suffix _P and _N.
It's possible to have a fourth data signal instead of clock as assembly option.

Pin assignment and function of both RJ45 connectors are shown in table below.

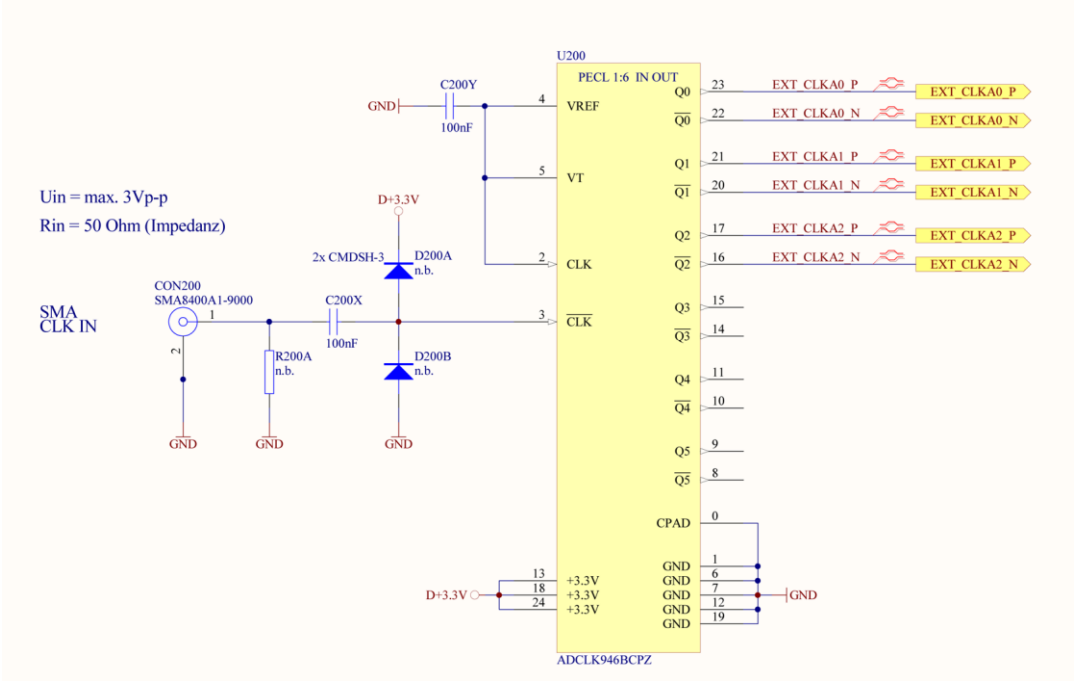
Pin	Signal Name	Function
1	**C_P** or **D0_P**	Clock or Data 0, positive signal of differential pair
2	**C_N** or **D0_N**	Clock or Data 0, negative signal of differential pair
3	**D1_P**	Data 1, positive signal of differential pair
4	**D1_N**	Data 1, negative signal of differential pair
5	**D2_P**	Data 2, positive signal of differential pair
6	**D2_N**	Data 2, negative signal of differential pair
7	**D3_P**	Data 3, positive signal of differential pair
8	**D3_N**	Data 3, negative signal of differential pair

Pin 1 and 2 signals are configured as Clock by default

Note:
If pins 1/2 are configured for D0 two limitations exists.
Signal D0 at Input connector is not available if system clock 2 option will be used.
Signal D0 at Output connector is not available if White Rabbit option will be used.
Signal name "Data" is referred to bits "Test Output" and "Input" in chapter 7.5.9 RJ45 Connector Input/Output Control register.

5.2 SMA Clock Input

The front panel SMA clock input is designed to accept a maximum peak to peak signal level of 3V into 50 Ohms. The clock input signal is coupled to the internal logic via a capacitor. The schematic of the input stage is shown below.



5.3 SFP Card Cage

The dual card cage is a zSFP+ type for best EMI protection and can host two SFP or SFP+ link media.

They can be enabled or disabled in the sis8300KU_top.vhd VHDL code as shown below :

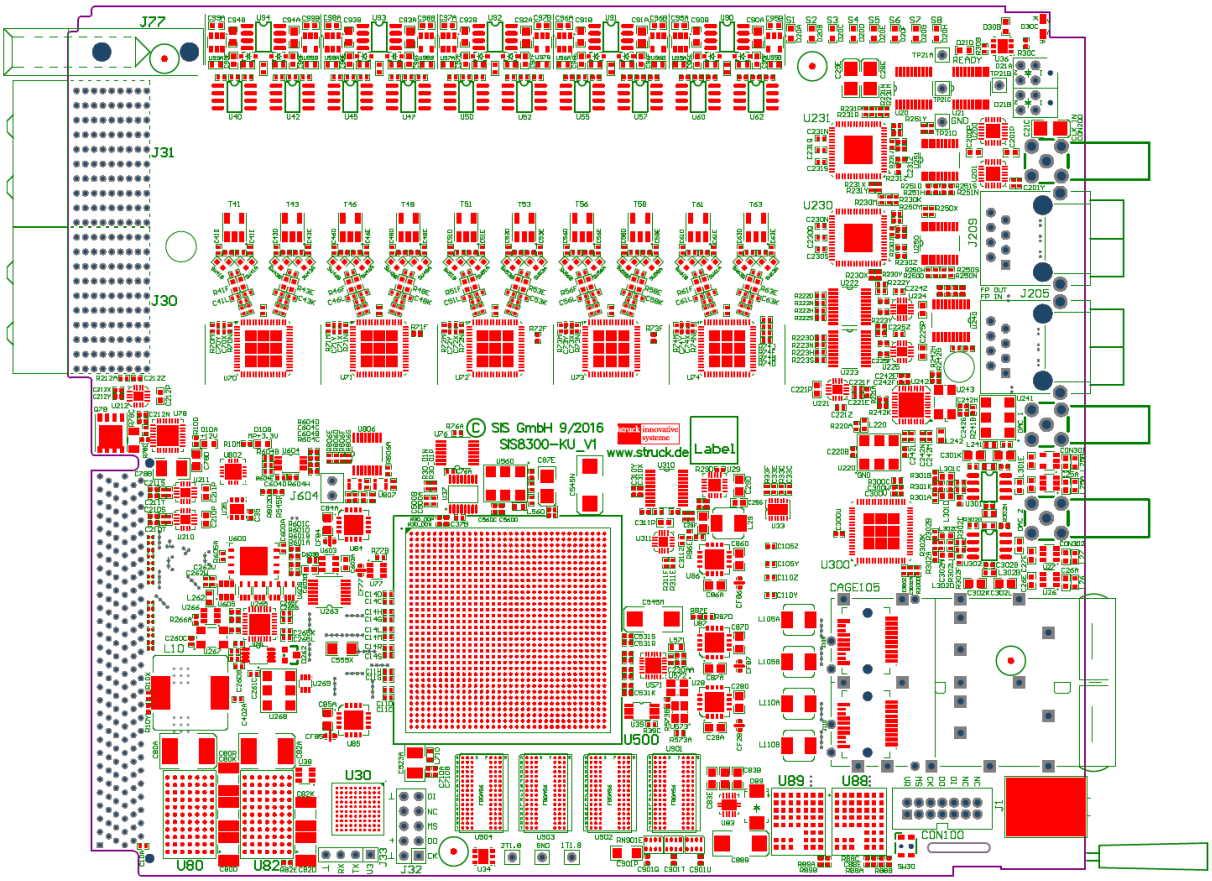
```
DUAL_OPTICAL_INTERFACE_EN       : integer := 1 ; --
```

Communication is handled through registers 0x14 to 0x19 (refer to the VHDL code)

Note: Dust covers (TE Connectivity part number 1367147-3) should be installed when no SFP media are in place.

6 Board Layout

A print of the silk screen of the component side is shown below.



Connector types:
The used connectors are listed in the table below.

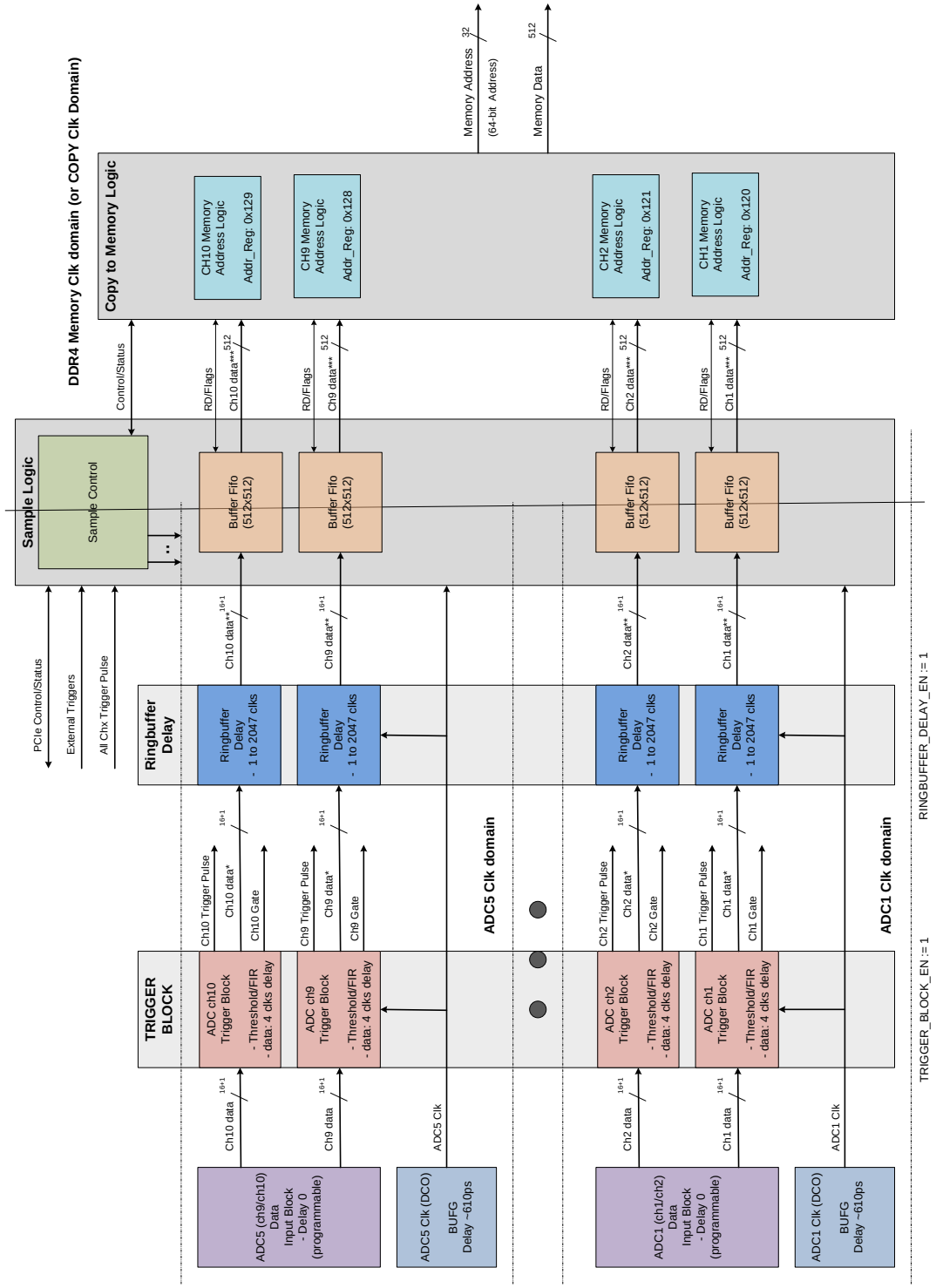
Designator	Function	Manufacturer	Part Number
CON100	JTAG	Molex	87831-1420
CON200	Clock In	JYEBAO	SMA8400A1-9000
CON301	DAC 1 Out	JYEBAO	SMA8400A1-9000
CON302	DAC 2 Out	JYEBAO	SMA8400A1-9000
CAGE105	SFP Cage, 2 Ports	TE Connectivity	1761014-1
J10	AdvancedMC	JBT	16211701303000
J30	Zone 3	ERNI	973028
J31	Zone 3	ERNI	973028
J32	JTAG ATxmega128	SAMTEC	HTSW-105-26-G-D
J33	ATxmega128 Debug	Fischer	MK1/4/G
J77	µRTM Keying	TYCO	5120913-3 (*)
J209	Data & Clock Out	ERNI	133268
J205	Data & Clock In	ERNI	133268
J604	Watchdog	SAMTEC	HTSW-102-26-G-S

Note (*): The used Key may depend on the hardware configuration of the SIS8300-KU

7 Firmware Description

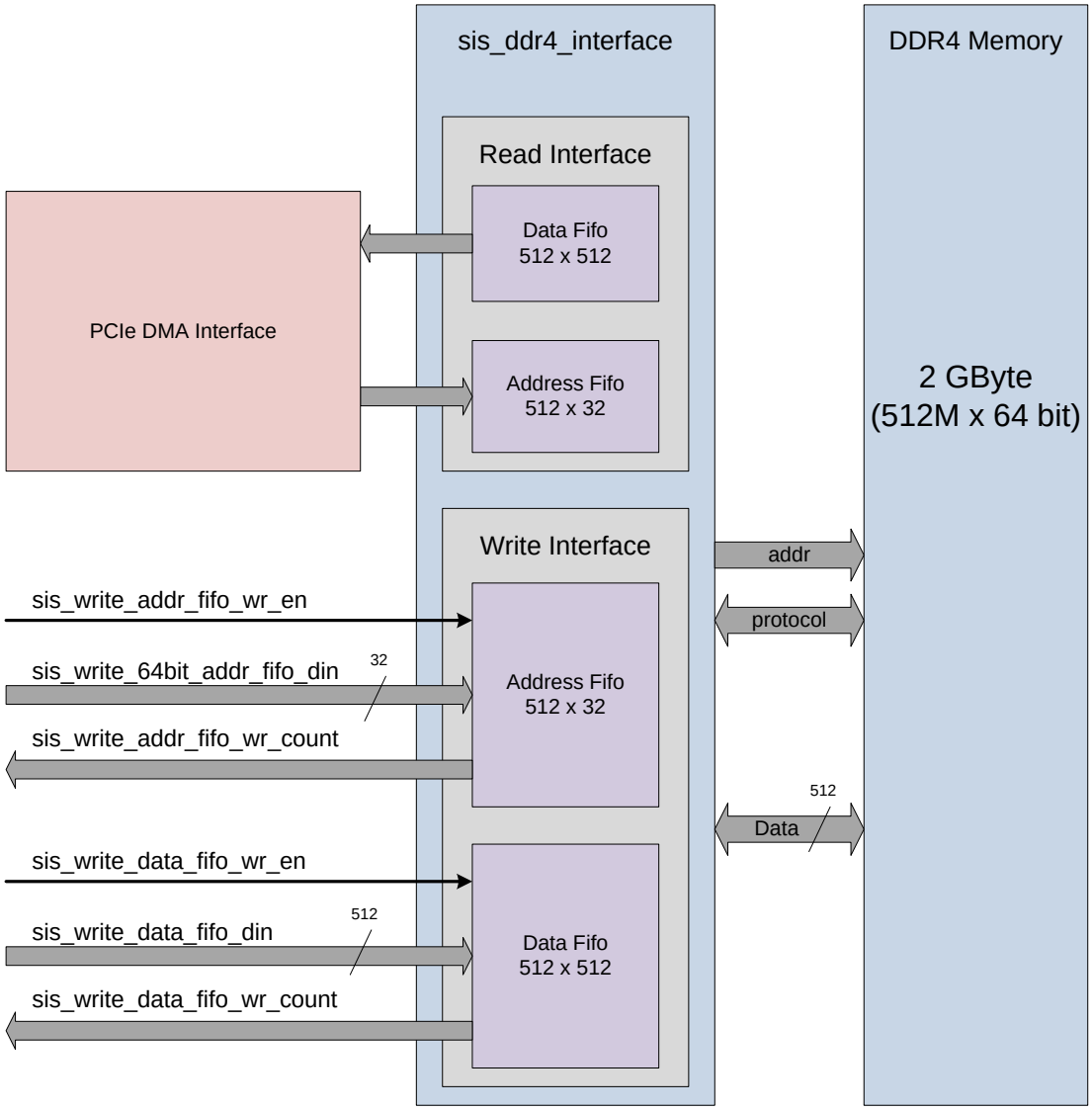
7.1 ADC Sample Logic

The block diagram shows the ADC data handling. Each ADC channel has its own Memory Address Control Logic.



7.2 Memory Interface

A block diagram of the DDR4 memory controller is shown below.



Note: the data width has changed from 256 bit to 512 bit with the SIS8300-KU.

7.2.1 Memory Write Interface

The Write Interface consists of the following signals:

```
write_fifo_wr_clk : in std_logic;
-- data: write fifo
write_data_fifo_wr_en : in std_logic;
write_data_fifo_din : in std_logic_vector(511 downto 0);
write_data_fifo_wr_count : out std_logic_vector(9 downto 0);

-- address: write fifo
write_addr_fifo_wr_en : in std_logic;
write_addr_fifo_din : in std_logic_vector(31 downto 0);
write_addr_fifo_wr_count : out std_logic_vector(9 downto 0);
```

A write cycle to the memory consists of one write command to the Address Fifo and one write command to the Data Fifo.

One write command to the Address FIFO:
a valid "sis_write_addr_fifo_wr_en" signal over one clock period (sis_write_fifo_wr_clk) along with "write_addr_fifo_din" (marked as "sis_write_64bit_addr_fifo_din" in the blockdiagram).

One write commands to the Data FIFO:
a valid "sis_write_data_fifo_wr_en" signal over one clock periods (sis_write_fifo_wr_clk) along with "sis_write_data_fifo_din".

When issuing a write command to the Address Fifo, the write command to the Data Fifo must be issued no more than zero clock cycle later.

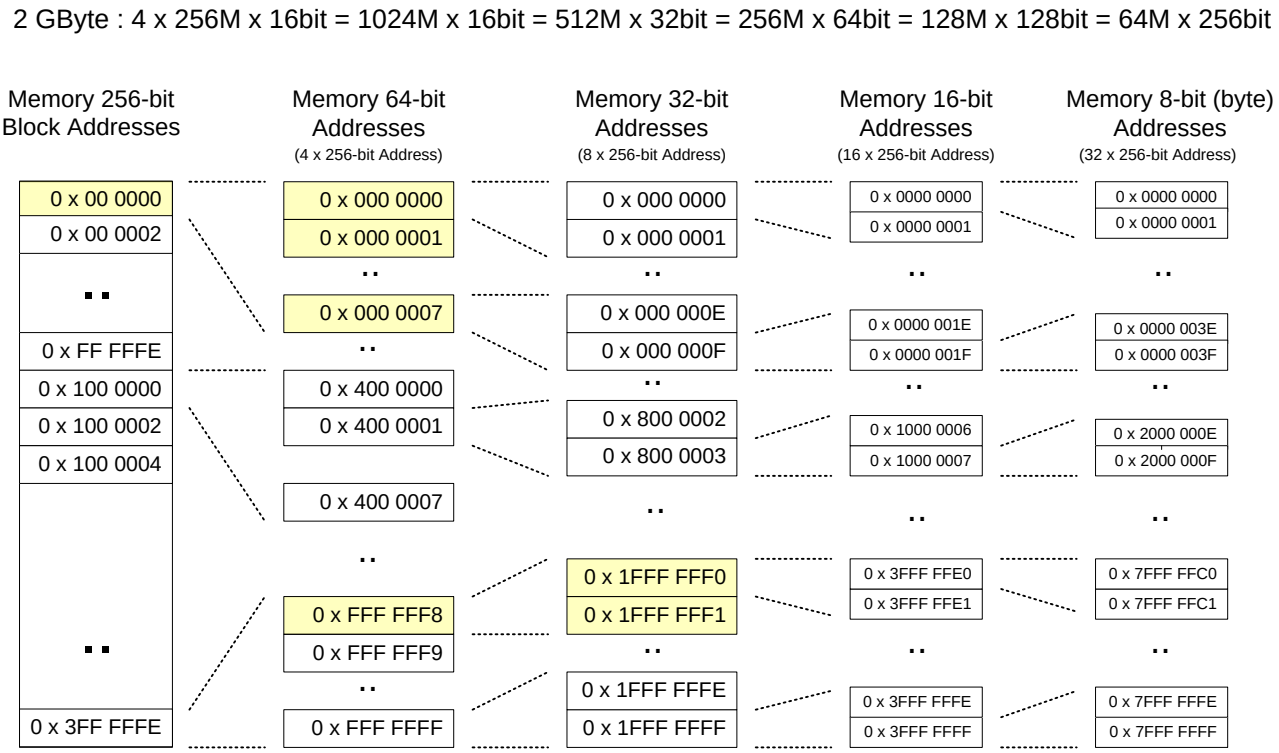
It is only allowed to write to the Address-FIFO, if "sis_write_addr_fifo_wr_count" is lower than X"1FF" (not full).
It is only allowed to write to the Data-FIFO, if "sis_write_data_fifo_wr_count" is lower than X"1FF" (not full).

The Memory Controller writes 512 bits (8 x 64 bits) to memory with one "write cycle". Therefore the lower 3 address bits of the written 64-bit address must be 0 and the "next address" will be incremented by 8.

Note again: the data width has changed from 256 bit to 512 bit with the SIS8300-KU and therefore the logic has to write in packages of 512 bits (64 Bytes) into the memory.

7.3 Memory buffer

The structure of the memory buffer with **2 GByte** (i.e. 4 x 4 GBit memory chips) is illustrated below.



"Block Addresses" are defined as 256-bit addresses and they are used to program the Sample Start Address (see Memory Sample Start Block Address register) and the Sample Length (see Sample Block Length register).

Note again: the data width has changed from 256 bit to 512 bit with the SIS8300-KU and therefore only even "Block addresses" and even "Block length" values are valid.

7.4 Address Map

Following 32-bit addresses are implemented

Offset	Access	Function	Note
0x00	R	Module Identifier/Firmware Version register	
0x01	R	Serial number register	
0x02	R/W	reserved	
0x03	R/W	reserved	
0x04	R/W	User Control/Status register (JK)	
0x05	R	Firmware Options register	
0x06	R/W	ADC Temperature Sensor interface register	
0x07	R	PCIe Status register	1
0x10	R/W	ADC Acquisition Control/Status register	
0x11	R/W	ADC Sample Control register	
0x12	R/W	MLVDS Input/Output Control register	
0x13	R/W	RJ45 Connector Input/Output Control register	1
0x14	R/W	SFP1 Link Control/Status register	1
0x15	R/W	SFP2 Link Control/Status register	1
0x16	R/W	Port 12 Link Control/Status register	1
0x17	R/W	Port 13 Link Control/Status register	1
0x18	R/W	Port 14 Link Control/Status register	1
0x19	R/W	Port 15 Link Control/Status register	1
0x20	R/W	DAC Trigger control register	
0x40	R/W	Clock Distribution Multiplexer control register	
0x41	R/W	Clock Distribution IC AD9510 SPI interface register	
0x42	R/W	Clock Multiplier IC SI5326 SPI interface register	
0x43	R/W	Clock Synthesizer Si5338A for MGT clocks	
0x44	R/W	FPGA Boot SPI Flash interface	
0x45	R/W	DAC Control register	2
0x46	R/W	DAC Data register	
0x47	R/W	RTM I2C interface register	
0x48	R/W	ADC SPI Interface register	
0x49	R/W	ADC Input Tap delay register	2
0x4E	R/W	DAC Trigger and DAC_CLK Prescaler setup register	
0x4F	R/W	DAC RAM endpoint register	
0xFF	W	Bit 0 = 1: Master Reset (reset all registers)	

Notes 1: new with SIS8300-KU
2: changed from SIS8300L/L2 to SIS8300-KU

Offset	Access	Function	Note
0x100	R/W	ADC ch1 Trigger Setup register	
0x101	R/W	ADC ch2 Trigger Setup register	
. .		. .	
. .		. .	
0x109	R/W	ADC ch10 Trigger Setup register	
0x110	R/W	ADC ch1 Trigger Threshold register	
0x111	R/W	ADC ch2 Trigger Threshold register	
. .		. .	
0x119	R/W	ADC ch10 Trigger Threshold register	
0x120	R/W	ADC ch1 Memory Sample Start Block Address / Actual Block Address register	2
0x121	R/W	ADC ch2 Memory Sample Start Block Address / Actual Block Address register	2
. .		. .	
0x129	R/W	ADC ch10 Memory Sample Start Block Address / Actual Block Address register	2
0x12A	R/W	ADC chx Sample Block Length register	2
0x12B	R/W	ADC chx Ringbuffer Delay register (0 to 2046)	
0x12F	R/W	SIS8900 RTM LVDS Test Input/Output Control register	
0x200	R/W	DMA_READ_DST_ADR_LO32	
0x201	R/W	DMA_READ_DST_ADR_HI32	
0x202	R/W	DMA_READ_SRC_ADR_LO32	
0x203	R/W	DMA_READ_LEN	
0x204	R/W	DMA_READ_CTRL	
0x205	R/W	DMA Readout Sample byte swap control	
0x210	R/W	DMA_WRITE_SRC_ADR_LO32	
0x211	R/W	DMA_WRITE_SRC_ADR_HI32	
0x212	R/W	DMA_WRITE_DST_ADR_LO32	
0x213	R/W	DMA_WRITE_LEN	
0x214	R/W	DMA_WRITE_CTRL	
0x215	R/W	DMA_ PC2CARD_MAX_NOF_OUTSTANDING_REQUESTS	
0x216	R/W	DAQ Auto DMA Chain Control	
0x220	R/W	IRQ Enable	
0x221	R	IRQ Status	
0x222	W	IRQ Clear	
0x223	KA	IRQ Refresh	
0x230	R/W	MEMORY test Mode register	
0x231	R/W	RAM FIFO debug register	
0x400	R/W	Mapped out of register bank to top level. May be used for user defined register implementation. See Section 7.6.	
. . .			
. . .			
0x4FF	R/W		

7.5 Register description

7.5.1 Module Id. and Firmware Revision register

#define SIS8300_IDENTIFIER_VERSION_REG 0x0

This register holds the module identifier (SIS8303) and the firmware version and revision.

BIT	access	Name	Function
31-16 FFFF0000	RO	Module Identifier	0x8303
15-8 0000FF00	RO	Firmware Version	1..255
7-0 000000FF	RO	Firmware Revision	1..255

Example: The initial versions of the SIS8300-KU reads 0x83031001 and 0x83032001

Meaning of the several firmware versions:

- 8303**1**xxx: standard versions (for RTM8900)
- 8303**2**xxx: versions with µRTM-USER-I²C support on Zone 3 Connector J30 4a/b (DWCnnnn)

7.5.2 Serial Number register

#define SIS8300_SERIAL_NUMBER_REG 0x01

This register holds the Serial Number of the module.

BIT	access	Name	Function
31-16 FFFF0000	RO	reserved	
15-0 0000FFFF	RO	Serial Number	1..65535

7.5.3 User Control/Status register

#define SIS8300_USER_CONTROL_STATUS_REG 0x04

The control register is implemented as a selective J/K register, a specific function is enabled by writing a 1 into the set/enable bit, the function is disabled by writing a 1 into the clear/disable bit (which location is 16-bit higher in the register). An undefined toggle status will result from setting both the enable and disable bits for a specific function at the same time. The only function at this point in time is user LED on/off.
On read access the same register represents the status register.

Bit	write Function	read Function
31	Clear reserved 15 (*)	0
30	Clear reserved 14 (*)	0
29	Clear reserved 13 (*)	0
28	Clear reserved 12 (*)	0
27	Clear reserved 11 (*)	0
26	Clear reserved 10 (*)	0
25	Clear reserved 9 (*)	0
24	Clear reserved 8 (*)	0
23	Clear reserved 7 (*)	0
22	Clear reserved 6 (*)	0
21	Clear reserved 5 (*)	0
20	Clear reserved 4 (*)	0
19	Clear reserved 3 (*)	0
18	Clear reserved 2 (*)	0
17	Switch off LED test	0
16	Switch off user LED (*)	0
15	Set reserved 15	Status reserved 15
14	Set reserved 14	Status reserved 14
13	Set reserved 13	Status reserved 13
12	Set reserved 12	Status reserved 12
11	Set reserved 11	Status reserved 11
10	Set reserved 10	Status reserved 10
9	Set reserved 9	Status reserved 9
8	Set reserved 8	Status reserved 8
7	Set reserved 7	Status reserved 7
6	Set reserved 6	Status reserved 6
5	Set reserved 5	Status reserved 5
4	Set reserved 4	Status reserved 4
3	Set reserved 3	Status reserved 3
2	Set reserved 2	Status reserved 2
1	Switch on LED test	Status LED test
0	Switch on user LED	Status User LED (1=LED on, 0=LED off)

(*) denotes power up default setting

7.5.4 Firmware Options register

#define SIS8300_FIRMWARE_OPTIONS_REG 0x05

This register holds the information of the Xilinx firmware option features.

Bit	read Function
31	reserved
30	...
...	...
16	Hardware Version flag
15	reserved
14	reserved
13	reserved
12	reserved
11	reserved
10	reserved
9	reserved
8	Z3 Class Bit 1
7	Z3 Class Bit 0
6	DUAL_OPTICAL_INTERFACE_EN
5	none
4	QUAD_PORT12_13_14_15_INTERFACE_EN
3	RTM_ZONE3_I2C_EN
2	none
1	RINGBUFFER_DELAY_EN
0	TRIGGER_BLOCK_EN

Note: For more information about Z3 Class Bits, please refer to section 7.5.31.

7.5.5 ADC Temperature Sensor interface register

```
#define SIS8300_ADC_TEMP_I2C_REG                0x06
```

Bit	Write	read
31	unused	Write/Read Logic BUSY Flag
30	unused	0
29	unused	0
28	unused	0
27	unused	0
..	..	..
16	unused	0
15	unused	0
14	unused	0
13	Byte Read cycle	0
12	Byte Write cycle	0
11	Issue STOP condition	0
10	Issue REPEATSTART condition	0
9	Issue START condition	0
8	Master I²C ACK bit, written during reads	Device I²C ACK bit, read during writes
7	Write Data Bit 7 (MSB)	Read Data Bit 7 (MSB)
..	..	..
1	Write Data Bit 1	Read Data Bit 1
0	Write Data Bit 0 (LSB)	Read Data Bit 0 (LSB)

The power up default value is 0x0

7.5.6 Firmware Options register

#define SIS8300KU_PCIE_STATUS_REG 0x07

This register holds the information of the PCI Express Link.

Bit	read Function
31	reserved
30	...
...	...
13	reserved
12	reserved
11	0
10	PCI Express Link Max_Payload_Size bit 2
9	PCI Express Link Max_Payload_Size bit 1
8	PCI Express Link Max_Payload_Size bit 0
7	0
6	PCI Express Link negotiated Speed bit 2
5	PCI Express Link negotiated Speed bit 1
4	PCI Express Link negotiated Speed bit 0
3	PCI Express Link negotiated Width bit 3
2	PCI Express Link negotiated Width bit 2
1	PCI Express Link negotiated Width bit 1
0	PCI Express Link negotiated Width bit 0

PCI Express Link Max_Payload_Size

bits 2:0	Maximum payload size
000	128 bytes
001	256 bytes
010	512 bytes
011	1024 bytes
100	2048 bytes
101	4096 bytes

PCI Express Link negotiated Speed

bits 2:0	Negotiated Speed
001	Gen1: 2.5 GT/s
010	Gen2: 5 GT/s
100	Gen3: 8 GT/s

PCI Express Link negotiated Width

bits 2:0	Negotiated Width
0001	x1: 1 lane
0010	x2: 2 lanes
0100	x4: 4 lanes
1000	x8: 8 lanes (not possible)

7.5.7 ADC Acquisition Control/Status register

```
#define SIS8300_ACQUISITION_CONTROL_STATUS_REG        0x10
```

Bit	write	read
31		0
..		..
..		..
8		0
7		Status: DDR4 Memory Init OK
6		0
5		Status: internal Sample Logic Buffer FIFO Not Empty
4		Status: internal Sample Logic Busy
3		0
2	'1' : Disable Sampling (Reset Sample Logic)	0
1	'1' : Arm Sampling (Start with next trigger)	Status: Arm for trigger (Wait for trigger)
0	'1' : Start Sampling immediately (Arm and Start/Trigger)	Status: Sampling Busy

The power up default value is 0x80

7.5.8 ADC Sample Control register

```
#define SIS8300_SAMPLE_CONTROL_REG        0x11
```

ADC channels can be disabled from storing data to memory by setting the corresponding disable bit in this register.

Bit	write
31	
..	
..	
12	
11	Enable external Trigger
10	Enable internal Trigger
9	Disable Sampling Ch10
8	Disable Sampling Ch9
7	..
6	..
5	..
4	..
3	..
2	Disable Sampling Ch3
1	Disable Sampling Ch2
0	Disable Sampling Ch1

The power up default value is 0x0

7.5.9 MLVDS Input/Output Control register

```
#define SIS8300_MLVDS_IO_CONTROL_REG      0x12
```

Bit	Write	Read
31	Enable LVDS Output Bit 7	Enable LVDS Output Bit 7
30	Enable LVDS Output Bit 6	Enable LVDS Output Bit 6
..	..	..
25	Enable LVDS Output Bit 1	Enable LVDS Output Bit 1
24	Enable LVDS Output Bit 0	Enable LVDS Output Bit 0
23	LVDS Output Bit 7	LVDS Output Bit 7
22	LVDS Output Bit 6	LVDS Output Bit 6
..	..	..
17	LVDS Output Bit 1	LVDS Output Bit 1
16	LVDS Output Bit 0	LVDS Output Bit 0
15	LVDS Input 7 External Trigger Enable	LVDS Input 7 External Trigger Enable
14	LVDS Input 6 External Trigger Enable	LVDS Input 6 External Trigger Enable
..	..	..
9	LVDS Input 1 External Trigger Enable	LVDS Input 1 External Trigger Enable
8	LVDS Input 0 External Trigger Enable	LVDS Input 0 External Trigger Enable
7	LVDS Input 7 External Trigger falling edge	LVDS Input Bit 7
6	LVDS Input 6 External Trigger falling edge	LVDS Input Bit 6
..	..	..
1	LVDS Input 1 External Trigger falling edge	LVDS Input Bit 1
0	LVDS Input 0 External Trigger falling edge	LVDS Input Bit 0

Note: external trigger in signals are synchronized to the FPGA CLK05

The register related FPGA pins are connected via MLVDS transceivers to the RX/TX ports of AMC multi-point bus. Assignment of register bits to AMC ports is shown in the table below.

AMC Port	LVDS Bit
TX20	7
RX20	6
TX19	5
RX19	4
TX18	3
RX18	2
TX17	1
RX17	0

7.5.10 RJ45 Connector Input/Output Control register

```
#define SIS8300KU_RJ45_IO_CONTROL_REG          0x13
```

Bit	Write	Read
31	No function	0
30	No function	0
..		
21	No function	0
20	Test Output Enable (**)	Test Output Enable
19	Test Output 3 (*)	Test Output 3
18	Test Output 2 (*)	Test Output 2
17	Test Output 1 (*)	Test Output 1
16	Test Output 0 (*)(~)	Test Output 0
15	Input 3 External Trigger falling edge	Input 3 External Trigger falling edge
14	Input 2 External Trigger falling edge	Input 2 External Trigger falling edge
13	Input 1 External Trigger falling edge	Input 1 External Trigger falling edge
12	Input 0 External Trigger falling edge (#)	Input 0 External Trigger falling edge
11	Input 3 External Trigger Enable	Input 3 External Trigger Enable
10	Input 2 External Trigger Enable	Input 2 External Trigger Enable
9	Input 1 External Trigger Enable	Input 1 External Trigger Enable
8	Input 0 External Trigger Enable (#)	Input 0 External Trigger Enable
7	No function	0
6	No function	0
5	No function	0
4	No function	0
3	No function	Input 3
2	No function	Input 2
1	No function	Input 1
0	No function	Input 0 (#)

(*): only if “Test Output Enable” = 1
(**): if “Test Output Enable” = 0 then RJ45 Output(1) = adc chx trigger pulse
(#): not relevant if RJ45 Input connector clock will be used (default state)
(~): not relevant if RJ45 Output connector clock will be used (default state)

Note:
External trigger in signals are synchronized to the FPGA CLK05.

7.5.11 SFP1 Link Control/Status register

```
#define SIS8300KU_SFP1_LINK_CONTROL_STATUS_REG          0x14
```

Bit	Write	Read
31	SFP1 and SFP2 GTH reset pulse	0
30		DMA Output Fifo full Flag
29		DMA Output Fifo halffull Flag
28		DMA Output Fifo empty Flag
27		0
26		0
25		DMA Input Fifo read count D9
24		0
23		0
..		
17		0
16		DMA Input Fifo read count D0
15	Transmit logic reset pulse	0
14	Receive logic reset pulse	0
13	Transmit logic enable	Status of Transmit logic enable
12	Receive logic enable	Status of Receive logic enable
11		Receive Link Status lane up
10		Receive Link Status channel up
9		Receive Link Status soft error
8		Receive Link Status hard error
7		Receive Link Status lane up changed latch bit
6		Receive Link Status channel up changed latch bit
5		Receive Link Status soft error latch bit
4	Clear Latch bits pulse	Receive Link Status hard error latch bit
3		NOT_GT_PLL_LOCK bit
2		0
1		SFP1 present
0		Receive Link Status OK

NOT_GT_PLL_LOCK bit = 0: gt_pll_lock is set.

7.5.12 SFP2 Link Control/Status register

```
#define SIS8300KU_SFP2_LINK_CONTROL_STATUS_REG          0x15
```

Bit	Write	Read
31		0
30		DMA Output Fifo full Flag
29		DMA Output Fifo halffull Flag
28		DMA Output Fifo empty Flag
27		0
26		0
25		DMA Input Fifo read count D9
24		0
23		0
..		..
17		0
16		DMA Input Fifo read count D0
15	Transmit logic reset pulse	0
14	Receive logic reset pulse	0
13	Transmit logic enable	Status of Transmit logic enable
12	Receive logic enable	Status of Receive logic enable
11		Receive Link Status lane up
10		Receive Link Status channel up
9		Receive Link Status soft error
8		Receive Link Status hard error
7		Receive Link Status lane up changed latch bit
6		Receive Link Status channel up changed latch bit
5		Receive Link Status soft error latch bit
4	Clear Latch bits pulse	Receive Link Status hard error latch bit
3		NOT_GT_PLL_LOCK bit
2		0
1		SFP2 present
0		Receive Link Status OK

NOT_GT_PLL_LOCK bit = 0: gt_pll_lock is set.

7.5.13 Port12 Link Control/Status register

```
#define SIS8300KU_PORT12_LINK_CONTROL_STATUS_REG          0x16
```

Bit	Write	Read
31	Port12,13,14 and 15 GTH reset pulse	0
30		DMA Output Fifo full Flag
29		DMA Output Fifo halffull Flag
28		DMA Output Fifo empty Flag
27		0
26		0
25		DMA Input Fifo read count D9
24		0
23		0
..		..
17		0
16		DMA Input Fifo read count D0
15	Transmit logic reset pulse	0
14	Receive logic reset pulse	0
13	Transmit logic enable	Status of Transmit logic enable
12	Receive logic enable	Status of Receive logic enable
11		Receive Link Status lane up
10		Receive Link Status channel up
9		Receive Link Status soft error
8		Receive Link Status hard error
7		Receive Link Status lane up changed latch bit
6		Receive Link Status channel up changed latch bit
5		Receive Link Status soft error latch bit
4	Clear Latch bits pulse	Receive Link Status hard error latch bit
3		NOT_GT_PLL_LOCK bit
2		0
1		0
0		Receive Link Status OK

NOT_GT_PLL_LOCK bit = 0: gt_pll_lock is set.

7.5.14 Port13/14/15 Link Control/Status register

```
#define SIS8300KU_PORT13_LINK_CONTROL_STATUS_REG          0x17
#define SIS8300KU_PORT14_LINK_CONTROL_STATUS_REG          0x18
#define SIS8300KU_PORT15_LINK_CONTROL_STATUS_REG          0x19
```

Bit	Write	Read
31		0
30		DMA Output Fifo full Flag
29		DMA Output Fifo halffull Flag
28		DMA Output Fifo empty Flag
27		0
26		0
25		DMA Input Fifo read count D9
24		0
23		0
..		..
17		0
16		DMA Input Fifo read count D0
15	Transmit logic reset pulse	0
14	Receive logic reset pulse	0
13	Transmit logic enable	Status of Transmit logic enable
12	Receive logic enable	Status of Receive logic enable
11		Receive Link Status lane up
10		Receive Link Status channel up
9		Receive Link Status soft error
8		Receive Link Status hard error
7		Receive Link Status lane up changed latch bit
6		Receive Link Status channel up changed latch bit
5		Receive Link Status soft error latch bit
4	Clear Latch bits pulse	Receive Link Status hard error latch bit
3		NOT_GT_PLL_LOCK bit
2		0
1		0
0		Receive Link Status OK

NOT_GT_PLL_LOCK bit = 0: gt_pll_lock is set.

7.5.15 DAC Trigger Control register

```
#define SIS8300_DAC_TRIGGER_CONTROL_REG   0x20
```

Bit	Write	Read
31		0
...		..
4	Synch-Mode	Synch-Mode
7	reserved	
6	reserved	
5	DAC 2 - Stop converting	1: DAC 2 stopped
4	DAC 1 - Stop converting	1: DAC 1 stopped
3	DAC 2 - Arm converting (Start with next trigger)	Status: DAC 2 Arm for trigger (Wait for trigger)
2	DAC 2 - Start converting immediately (Arm and Start)	Status: DAC 2 Busy
1	DAC 1 - Arm converting (Start with next trigger)	Status: DAC 1 Arm for trigger (Wait for trigger)
0	DAC 1 - Start converting immediately (Arm and Start)	Status: DAC 1 Busy

Synch-Mode = 1: Both DACs are triggered synchronous with DAC1 “Start-logic”

7.5.16 Clock Distribution Multiplexer control register

```
#define SIS8300_CLOCK_DISTRIBUTION_MUX_REG      0x40
```

The SIS8300-KU has 5 IDT ICS853S057 clock multiplexer chips, which are labelled A to E in the clock distribution scheme in section 2.8. The multiplexer control register holds the two select bits for the 5 multiplexer chips as shown in the table below.
The assignment of the inputs to the resources (i.e. clock inputs) is listed in subsection 7.5.16.1.

BIT	**access**	**Name**	Function
31-12 FFFFF000	R/W	reserved	no
11-10 00000C00	R/W	MUXE_SEL	Multiplexer E select bits
9-8 00000300	R/W	MUXD_SEL	Multiplexer D select bits
7-6 000000C0	R/W	reserved	no
5-4 00000030	R/W	MUXC_SEL	Multiplexer C select bits
3-2 0000000C	R/W	MUXB_SEL	Multiplexer B select bits
1-0 00000003	R/W	MUXA_SEL	Multiplexer A select bits

7.5.16.1 Multiplexer A Input Signals:

U222 - Sel0 and Sel1 (MUX1A_SEL) = Multiplexer A select lines

Sel1	Sel0	Selected Input - Net Name	Clock source Description
0	0	RTMCLK_0	Clock 2 from µRTM card
0	1	TCLKB_0	Clock 2 (Telecom Clock B) from AMC Connector (Backplane)
1	0	TCLKA_0	Clock 1 (Telecom Clock A) from AMC Connector (Backplane)
1	1	OSC_CLK0	Onboard Clock chip (250MHz)

7.5.16.2 Multiplexer B Input Signals:

U223 - Sel0 and Sel1 (MUX1B_SEL) = Multiplexer B select lines

Sel1	Sel0	Selected Input - Net Name	Clock source Description
0	0	RTMCLK_1	Clock 2 from µRTM card
0	1	TCLKB_1	Clock 2 (Telecom Clock B) from AMC Connector (Backplane)
1	0	TCLKA_1	Clock 1 (Telecom Clock A) from AMC Connector (Backplane)
1	1	OSC_CLK1	Onboard Clock chip (250MHz)

7.5.16.3 Multiplexer C Input Signals:

U240 - Sel0 and Sel1 (MUXAB_SEL) = Multiplexer C select lines

Sel1	Sel0	Selected Input - Net Name	Clock source Description
0	0	EXT_CLKB0	Clock from RJ45 Connector “CI1-4 IN” (front panel)
0	1	EXT_CLKA0	Clock from SMA Connector “CLK IN” (front panel)
1	0	MUXA_CLK1	Multiplexer A Output Signal
1	1	MUXB_CLK1	Multiplexer B Output Signal

7.5.16.4 Multiplexer D Input Signals:

U250 - Sel0 and Sel1 (MUX2A_SEL) = Multiplexer D select lines

Sel1	Sel0	Selected Input - Net Name	Clock source Description
0	0	MUXA_CLK0	Multiplexer A Output Signal
0	1	MUL_CLK1	Clock Multiplier (U242) Output 2 Signal
1	0	EXT_CLKB1	Clock from RJ45 Connector “CI1-4 IN” (front panel)
1	1	EXT_CLKA1	Clock from SMA Connector “CLK IN” (front panel)

7.5.16.5 Multiplexer E Input Signals:

U251 - Sel0 and Sel1 (MUX2B_SEL) = Multiplexer E select lines

Sel1	Sel0	Selected Input - Net Name	Clock source Description
0	0	MUXB_CLK0	Multiplexer B Output Signal
0	1	MUL_CLK0	Clock Multiplier (U242) Output 1 Signal
1	0	EXT_CLKB2	Clock from RJ45 Connector “CI1-4 IN” (front panel)
1	1	EXT_CLKA2	Clock from SMA Connector “CLK IN” (front panel)

7.5.17 Clock Distribution AD9510 Serial Interface (SPI) interface register

```
#define SIS8300_AD9510_SPI_REG            0x41
```

The parameters of the Clock Distribution IC AD9510 chips can be configured with the SPI (serial Peripheral Interface).

Bit	Write	read	
31	Cmd Bit 1	Write/Read Logic BUSY Flag	
30	Cmd Bit 0		
29	Set “Function” Output Level	Status of Set “Function” Output Level	
28	Select “Function” synchronisation CLK	Status of Select “Function” synchronisation CLK	
...			
...			
25		Status AD9510 #2	
24	AD9510 #2 Select Bit	Status AD9510 #1	
23	Read Cycle Bit		RW CMD
22			
21			
20	Address Bit 12		
19	Address Bit 11		
...	...		
...	...		
12	Address Bit 4		
11	Address Bit 3		
10	Address Bit 2		
9	Address Bit 1		
8	Address Bit 0		
7	Write Data Bit 7 (MSB)	Read Data Bit 7 (MSB)	
6	Write Data Bit 6	Read Data Bit 6	
..	...	..	
1	Write Data Bit 1	Read Data Bit 1	
0	Write Data Bit 0 (LSB)	Read Data Bit 0 (LSB)	

The power up default value is 0x20000000

Command Bit (31:30) Explanation:

Cmd Bit 1	Cmd Bit 0	Command
0	0	No Function
0	1	R/W CMD
1	0	Function CMD Generates a pulse at the Function Input pin of the AD9510 which is synchronous to the selected clock. The clock selection is done via Bit 28 (Function Syn CLK). The actual function depends on the programming of the selected AD9510
1	1	Reserved

Select “Function” synchronisation CLK Bit (28) Explanation:

Bit 28	Clock Source
0	PCI Clock
1	FPGA CLK 69

Note:

1. enable READ by writing 0x90 to addr 0x0
2. and set Read Cycle Bit

Note:
Please refer to the SIS8300_AD9510_SPI_Setup routine as illustration and to the AD9510 documentation for details.

7.5.18 Clock Multiplier IC SI5326 SPI interface register

```
#define SIS8300_CLOCK_MULTIPLIER_SPI_REG        0x42
```

Several parameters of the Clock Multiplier SI5326 chip can be configured with the SPI (serial Peripheral Interface).
Please refer to the documentation of the SI5326 chip for details.

Bit	Write	read
31	Cmd Bit 1	Write/Read Logic BUSY Flag
30	Cmd Bit 0	Reset, Decrement or Increment Cmd BUSY Flag
29		
...		
17		Si53xx LOL Status
16		Si53xx INT_C1B Status
15	Instruction Byte Bit 7	
..		
..		
8	Instruction Byte Bit 0	
7	Address/Data Byte Bit 7	Read Data Bit 7 (MSB)
..	..	..
1	Address/Data Byte Bit 1	Read Data Bit 1
0	Address/Data Byte Bit 0	Read Data Bit 0 (LSB)

The power up default value is 0x0

Cmd Bit 1	Cmd Bit 0	Command
0	0	Execute SPI Write/Read Cmd
0	1	Reset Cmd
1	0	Decrement Cmd
1	1	Increment Cmd

Reset Cmd: generates an 1us reset pulse

Decrement Cmd: generates an 1us Skew Decrement pulse
Increment Cmd: generates an 1us Skew Increment pulse

Note: INC/DEC Time between consecutive pulses must be greater than 16ms !

The SI5326 chip has the two control pins called INC and DEC. They can be used to adjust input to output device skew directly rather than by sending commands.
This feature is currently not implemented in the FPGA design but planned as future extension.

7.5.19 Clock Synthesizer IC Si5338A I²C interface register

```
#define SIS8300_MGTCLK_SYNTH_I2C_REG      0x43
```

Several parameters of the Clock Synthesizer SI5338A chip can be configured with the I²C Interface.
Please refer to the documentation of the SI5338A chip for details.

Bit	Write	read
31	unused	Write/Read Logic BUSY Flag
30	unused	0
29	unused	0
28	unused	0
27	unused	0
..	..	..
16	unused	0
15	unused	0
14	unused	0
13	Byte Read cycle	0
12	Byte Write cycle	0
11	Issue STOP condition	0
10	Issue REPEATSTART condition	0
9	Issue START condition	0
8	Master I²C ACK bit, written during reads	Device I²C ACK bit, read during writes
7	Write Data Bit 7 (MSB)	Read Data Bit 7 (MSB)
..	..	..
1	Write Data Bit 1	Read Data Bit 1
0	Write Data Bit 0 (LSB)	Read Data Bit 0 (LSB)

The power up default value is 0x0

7.5.20 FPGA Boot SPI Flash interface

```
#define SIS8300_SPI_FLASH_REG       0x44
```

Bit	Write	read
31	Clear “Dout/CS/CCLK Output Enable” (*)	SPI logic busy
30	Clear “Flash chip select” (*)	0
29	Clear “Block mode” (*)	0
28	unused	0
27	unused	0
26	Clear “Read Block mode” (*)	0
25	unused	0
24	unused	0
23	unused	0
..	..	0
17	unused	0
16	unused	0
15	Set “Dout/CS/CCLK Output Enable”	Status “Dout/CS/CCLK Output Enable”
14	Set “Flash chip select” (CS)	Status “Flash chip select”
13	Set “Block mode”	Status “Block mode”
12	“Start” command	0
11	“Write buffer fifo” command	0
10	Set “Read Block mode”	Status “Read Block mode”
9	“Read buffer fifo” command	0
8	unused	0
7	Write Data Bit 7 (MSB)	Read Data Bit 7 (MSB)
..	..	..
1	Write Data Bit 1	Read Data Bit 1
0	Write Data Bit 0 (LSB)	Read Data Bit 0 (LSB)

(*) denotes power up default setting

Please refer to the library …\lib\libSIS830x\libSIS830x.so (`sis830x_spiFlashUpdateFirmware`) and the program \sis8300ku_tests\flash\main.c.

For more information about flashing SPI EEPROM, please refer to section 2.4 and 9.

7.5.21 DAC Control register

```
#define SIS8300_DAC_CONTROL_REG     0x45
```

Bit	write	read
31	Tap Delay Write Pulse	
30	Tap Delay value bit 8	
29	Tap Delay value bit 7	
28	Tap Delay value bit 6	
27	Tap Delay value bit 5	
26	Tap Delay value bit 4	
25	Tap Delay value bit 3	
24	Tap Delay value bit 2	
23	Tap Delay value bit 1	
22	Tap Delay value bit 0	
21	DAC 2 Wrap select	
20	DAC 1 Wrap select	
19		
18	FPGA CLK select	FPGA CLK: DAC Output Update Rate 0: 125MHz (ADC1 clock) 1: 250MHz (2 * ADC1 clock)
17	DAC Clock Multiplexer select bit 1	DAC Clock Multiplexer select bit 1
16	DAC Clock Multiplexer select bit 0	DAC Clock Multiplexer select bit 0
15		
...		
12		
11		
10		
9		
8	DAC DCM Reset pulse	
7		
6		
5	Power Down/Up	0: power down, 1: power up
4	TORB Two's-Complement/Binary Select	0: binary, 1: Two's complement
..		
1	Test Mode Bit 1	
0	Test Mode Bit 0	

The power up default value is 0x0

DAC Clock Multiplexer table

select bit 1	select bit 0	DAC clock source
0	0	FPGA clock
0	1	Clock Divider AD9510 #2 Out 0
1	0	Clock Divider AD9510 #1 Out 0
1	1	RTM Clk4

Test mode bit function table

Test Mode Bit 1	Test Mode Bit 0	DAC Test Mode
0	0	Data from DAC Data register
0	1	Ramp Test Mode
1	0	ADC1/ADC2 -> DAC1/DAC2
1	1	Data from DAC RAM

Note 1: ADC 1 Clock is used as DAC clock (refer chapter: 7.5.26)

Note 2: The RAM store up to 64k (**65536)** values per DAC.
The '*DMA write enabled*' Bit must set to enable the direct memory write access to the DAC RAM (refer chapter 7.5.39 and onwards for future information's). The 'DMA Data' format corresponds to the 'DAC Data register' format (refer chapter 7.5.22).

Note 3: It can be necessary to setup the *'tap delay'* to composite a device dependent asynchronicity between the clock path and the data path of the DAC. An indicator for an excessive shift is an disturbed DAC output signal.
With the *'tap delay*' value it is possible to shift the data path relative to the clock signal to resynchronize the two paths.

7.5.22 DAC Data register

```
#define SIS8300_DAC_DATA_REG        0x46
```

Bit	write
31	DAC2 Data 15
...	
...	
16	DAC2 Data 0
15	DAC1 Data 15
...	
...	
0	DAC1 Data 0

The power up default value is 0x0, data=0 → +1 V, data=0xFFFF → - 1V output (with TORB=1, i.e. in Two's complement mode)

Note: The default DAC range on the DAC Out SMA connector is -1V,…,+1V into a 50 Ω load

7.5.23 RTM I2C interface register

```
#define SIS8300_RTM_I2C_BUS_REG     0x47
```

Rear Transition Modules (µRTMs) like the DWC8VM1 or DWC8300 have components that are configured and/or read out by an I²C interface over the Zone 3 connector.
This register furnishes the interface for I²C read and write access.
It is implemented in firmware versions V2xxx.
A software example can be found on the Struck product DVD under:
sisdvd_xxxxxx\sis8xxx and DWC\sis8300L\software\tests\rtm_i2c_test
Please refer to the documentation of the respective RTM for details.

Bit	Write	read
31	unused	Write/Read Logic BUSY Flag
30	unused	0
29	unused	0
28	unused	0
27	unused	0
..		
16	unused	0
15	unused	0
14	unused	0
13	Byte Read cycle	0
12	Byte Write cycle	0
11	Issue STOP condition	0
10	Issue REPEATSTART condition	0
9	Issue START condition	0
8	Master I²C ACK bit, written during reads	Device I²C ACK bit, read during writes
7	Write Data Bit 7 (MSB)	Read Data Bit 7 (MSB)
..		
1	Write Data Bit 1	Read Data Bit 1
0	Write Data Bit 0 (LSB)	Read Data Bit 0 (LSB)

The power up default value is 0x0

7.5.24 ADC Serial Interface (SPI) interface register

```
#define SIS8300_ADC_SERIAL_INTERFACE_REG        0x48
```

Several parameters of the ADC AD9268 chip can be configured with the SPI (serial Peripheral Interface).
Please refer to the documentation of the ADC AD9268 chip for details.

Bit	write	read
31	ADC Synch cmd	Write/Read Logic BUSY Flag
..		
26	ADC Select Mux Bit 2	
25	ADC Select Mux Bit 1	
24	ADC Select Mux Bit 0	
23	Read Cmd	
22	not used	
21	not used	
20	not used	
19	not used	
14	not used	
13	Address Bit 5	
12	Address Bit 4	
11	Address Bit 3	
10	Address Bit 2	
9	Address Bit 1	
8	Address Bit 0	
7	Write Data Bit 7 (MSB)	Read Data Bit 7 (MSB)
6	Write Data Bit 6	Read Data Bit 6
..		
1	Write Data Bit 1	Read Data Bit 1
0	Write Data Bit 0 (LSB)	Read Data Bit 0 (LSB)

The power up default value is 0x0

ADC Synch Cmd : generates a synch pulse with the AD9510 #1 FPGA clock

7.5.25 ADC Input Tap delay register

```
#define SIS8300_ADC_INPUT_TAP_DELAY       0x49
```

The ADC input tap delay register is used to adjust the FPGA data strobe timing.

Bit	31-13	12	11	10	9	8	7-0
Function	None	ADC 9/10 Select	ADC 7/8 Select	ADC 5/6 Select	ADC 3/4 Select	ADC 1/2 Select	Write Tap delay value

Bit	write	read
31	reserved	0
...		
28	reserved	
27	reserved	0
26	ADC Read Select Mux Bit 2	0
25	ADC Read Select Mux Bit 1	0
24	ADC Read Select Mux Bit 0	0
23	reserved	Read Tap delay value Bit 7
22	reserved	Read Tap delay value Bit 6
..	..	
17	reserved	Read Tap delay value Bit 1
16	reserved	Read Tap delay value Bit 0
15	reserved	0
14	reserved	0
13	reserved	0
12	ADC 9/10 Write Select Bit	
11	ADC 7/8 Write Select Bit	
10	ADC 5/6 Write Select Bit	
9	ADC 3/4 Write Select Bit	
8	ADC 1/2 Write Select Bit	
7	Write Tap delay value Bit 7	Write Tap delay value Bit 7
6	Write Tap delay value Bit 6	Write Tap delay value Bit 6
..	..	..
1	Write Tap delay value Bit 1	Write Tap delay value Bit 1
0	Write Tap delay value Bit 0	Write Tap delay value Bit 0

The Write Tap delay value bits 7 downto 0 corresponds to IDELAY Tap delay value bits 8 downto 1 of the IDLEAY3 primitive of the Ultrascale (bit 0 is set 0).

No Tap delay is necessary with sample frequencies up to 125 MHz .

7.5.26 DAC Trigger and DAC_CLK prescaler setup register

```
#define SIS8300_DAC_TRIGGER_PRECLK_REG          0x4E
```

The DAC uses the same clock source like the ADC 1. With the DAC CLK prescaler it is possible to define a different clock period based on the ADC clock. The prescaler are set up by the divider Bits:

$$DAC\_CLK = \frac{SOURCE\_CLK}{divider + 1}$$

(A divider value of one generates a twice period time of the source)

The '*ADC trigger select Bits*' defines the internal trigger source:
0h = ADC_1, ..., 9h = ADC_10

Bit	write	read
31	DAC 2 CLK divider Bit 7	DAC 2 CLK divider Bit 7
...	...	...
...	...	...
24	DAC 2 CLK divider Bit 0	DAC 2 CLK divider Bit 0
23	DAC 1 CLK divider Bit 7	DAC 1 CLK divider Bit 7
...	...	...
...	...	...
16	DAC 1 CLK divider Bit 0	DAC 1 CLK divider Bit 0
15	-	0
14	-	0
13	DAC 2 Enable external Trigger	DAC 2 external Trigger enabled
12	DAC 2 Enable internal Trigger	DAC 2 internal Trigger enabled
11	DAC 2 ADC trigger select Bit 3	DAC 2 ADC trigger select Bit 3
10	DAC 2 ADC trigger select Bit 2	DAC 2 ADC trigger select Bit 2
9	DAC 2 ADC trigger select Bit 1	DAC 2 ADC trigger select Bit 1
8	DAC 2 ADC trigger select Bit 0	DAC 2 ADC trigger select Bit 0
7	-	0
6	-	0
5	DAC 1 Enable external Trigger	DAC 1 external Trigger enabled
4	DAC 1 Enable internal Trigger	DAC 1 internal Trigger enabled
3	DAC 1 ADC trigger select Bit 3	DAC 1 ADC trigger select Bit 3
2	DAC 1 ADC trigger select Bit 2	DAC 1 ADC trigger select Bit 2
1	DAC 1 ADC trigger select Bit 1	DAC 1 ADC trigger select Bit 1
0	DAC 1 ADC trigger select Bit 0	DAC 1 ADC trigger select Bit 0

The power up default value is 0x0

7.5.27 DAC RAM endpoint register

```
#define SIS8300_DAC_DATA_ENDP_REG              0x4F
```

This register defines the last element in the DAC RAM and mark the value at the RAM read logic 'wrap around' to the first RAM element (if the '*Wrap select*' Bit set at the DAC control register).
If *N* DAC values are written into the DAC RAM, the last value is placed on address *N-1*.

Bit	write	read
31	DAC 2 RAM endpoint Bit 15	DAC 2 RAM endpoint Bit 15
30	DAC 2 RAM endpoint Bit 14	DAC 2 RAM endpoint Bit 14
...	...	...
...	...	...
17	DAC 2 RAM endpoint Bit 1	DAC 2 RAM endpoint Bit 1
16	DAC 2 RAM endpoint Bit 0	DAC 2 RAM endpoint Bit 0
15	DAC 1 RAM endpoint Bit 15	DAC 1 RAM endpoint Bit 15
14	DAC 1 RAM endpoint Bit 14	DAC 1 RAM endpoint Bit 14
...	...	...
...	...	...
1	DAC 1 RAM endpoint Bit 1	DAC 1 RAM endpoint Bit 1
0	DAC 1 RAM endpoint Bit 0	DAC 1 RAM endpoint Bit 0

The power up default value is 0x0

7.5.28 Trigger registers

The Trigger Block contains Logic to generate internal triggers (only implemented if the Firmware Option register bit TRIGGER_BLOCK_EN = 1).
Two types are implemented: A “threshold trigger” and a “FIR trigger”.

7.5.28.1 Trigger setup registers

```
#define SIS8300_TRIGGER_SETUP_CH1_REG          0x100
..
#define SIS8300_TRIGGER_SETUP_CH10_REG         0x109
```

These read/write registers hold the 8-bit wide trigger pulse length (in sample clocks), the Peaking and Gap Time of the trapezoidal FIR filter.
(Gap Time = SumG Time – Peaking Time)

Bit	Function	
31	Reserved	
..	..	
26	Enable Trigger	
25	GT trigger condition	
24	FIR Trigger Mode (0: Threshold Trigger; 1: FIR Trigger)	
23	Puls Length bit 7	Trigger Pulse Length
22	Puls Length bit 6	
21	Puls Length bit 5	
20	Puls Length bit 4	
19	Puls Length bit 3	
18	Puls Length bit 2	
17	Puls Length bit 1	
16	Puls Length bit 0	
15	reserved	SumG time (only FIR trigger) (time between both sums)
14	reserved	
13	reserved	
12	SumG bit 4	
11	SumG bit 3	
10	SumG bit 2	
9	SumG bit 1	
8	SumG bit 0	
7	reserved	Peaking time P (only FIR trigger) $\sum_{i=x}^{x+P} Si$
6	reserved	
5	reserved	
4	P bit 4	
3	P bit 3	
2	P bit 2	
1	P bit 1	
0	P bit 0	

The power up default value reads 0x 00000000

Si: Sum of ADC input sample stream from x to x+P
P: Peaking time (number of values to sum)
SumG: SumGap time (distance in clock ticks of the two running sums)

The maximum SumG time: 16 (clocks)
The minimun SumG time: 1 (clocks)
Values > 16 will be set to 16
Value = 0 will be set to 1

The maximum Peaking time: 16 (clocks)
The minimun Peaking time: 1 (clocks)
Values > 16 will be set to 16
Value = 0 will be set to 1

7.5.28.2 Trigger Threshold registers

```
#define SIS8300_TRIGGER_THRESHOLD_CH1_REG            0x110
..
#define SIS8300_TRIGGER_THRESHOLD_CH10_REG           0x119
```

These read/write registers hold the threshold values for the 10 ADC channels.

7.5.28.2.1 Trigger Threshold

FIR Trigger Mode = 0

Bit	31-16	15-0
Function	Threshold value OFF	Threshold value ON

default after Reset: 0x0

A trigger output pulse is generated on two conditions:

- GT is set (GT) in trigger setup register:
  the trigger Out pulse will be issued if the actual sampled ADC value **goes** above the threshold value ON **and** OFF. A new Trigger Out Pulse will be suppressed until the ADC value **goes** below the threshold value OFF.

- GT is cleared (LT) in trigger setup register:
  the trigger Out pulse will be issued if the actual sampled ADC value **goes** below the threshold value ON **and** OFF. A new trigger Out pulse will be suppressed until the ADC value **goes** above the threshold value OFF.

  the trigger Out pulse will be issued if the actual sampled ADC value **goes** below the threshold value.

GT: greater than
LT: lower than

7.5.28.2.2 FIR Trigger Threshold

FIR Trigger Mode = 1

Bit	31-20	19-0
Function	None	**Trapezoidal t**hreshold value

default after Reset: 0x0

The value of the Sum (trapezoidal value) depends on the peaking time P. Therefore the selection of the value of the Trapezoidal threshold depends on P also.

Trapezoidal value calculation:
Trapezoidal value = (SUM2 – SUM1)

Where

$$SUM1 = \sum_{i=x}^{x+P} Si$$

$$SUM2 = \sum_{j=x+sumG}^{x+P+sumG} Sj$$

The FIR filter logic generates the Trapezoidal by subtraction of the two running sums. This implies, that the internal value of the trapezoid is on average 0.

A trigger output pulse is generated:

- GT is set (GT):
  the Trigger Out Pulse will be issued if the actual trapezoidal value **goes** above the programmable trapezoidal threshold value

- GT is cleared (LT):
  the Trigger Out Pulse will be issued if the actual trapezoidal value **goes** below the **negated** programmable trapezoidal threshold value

7.5.29 Memory Sample Start Block Address / Actual Sample Block Address registers

```
#define SIS8300_SAMPLE_START_ADDRESS_CH1_REG    0x120
..
#define SIS8300_SAMPLE_START_ADDRESS_CH10_REG   0x129
```

The **write function** to these registers defines the memory start block address.
The value is given in 256-bit blocks.
Please note, only even numbers are valid.

Write Function: ADC chx Memory Sample Start Block Address

Bit	31-26	25-1	0
	reserved	Memory Sample Start Block Address 25-1 (256-bit blocks)	Don't care

default after Reset: 0x0

Explanation (memory sample start block address)
The contents of the **sample memory start block address** register is assigned as memory data storage address with the arm command (key address arm sampling) or with the enable command (key address enable sampling).

The **read function** from these registers give the information of the actual sampling block address for the given ADC channel.
(at the moment: only valid if the logic is not busy!)

Read Function : ADC chx Actual Sample Block Address

Bit	31-26	25-0
	reserved	Actual Sample Block Address (in 256-bit Blocks) (16-bit word address x 16)

The value is given in 256-bit Blocks.

7.5.30 Sample Block Length register

```
#define SIS8300_SAMPLE_LENGTH_REG                0x12A
```

This register defines the number of sample blocks of each ADC channel.
The size of one sample block for each ADC channel is 256-bit (16 x 16-bit word).
Please note, only even numbers are valid.

Bit	31-26	25-1	0
	reserved	Sample Block Length 25-1 (256-bit blocks)	Don't care

default after reset: 0x0

Sample Block Length value	Number of samples of each channel (waveform length)
0x0 (0x1)	32
0x2 (0x3)	64
0x4 (0x5)	96
0x6 (0x7)	128
..	..
0x 3FF FFFE(*)	**0x 4000 0000 (1.073.741.824)**

* maximum block length of one channel only !

7.5.31 Ringbuffer Delay register

```
#define SIS8300_PRETRIGGER_DELAY_REG             0x12B
```

This register defines the number of pre trigger delay samples for all channels.
The maximum pre trigger delay value is 2046.

Bit	31-12	11-0
Function	reserved	Delay value

7.5.32 SIS8900 RTM LVDS Test Input/Output Control register

```
#define SIS8300_RTM_LVDS_IO_CONTROL_REG        0x12F
```

Bit	Write	Read
31	-	Z3 Class Bit 0
30	-	Z3 Class Bit 1
29	Enable RTM LVDS Output Bit 11 (*)	Enable RTM LVDS Output Bit D 11 (*)
28	Enable RTM LVDS Output Bit 10 (*)	Enable RTM LVDS Output Bit D 10 (*)
27	Enable RTM LVDS Output Bit 9 (*,#)	Enable RTM LVDS Output Bit D 9 (*,#)
26	Enable RTM LVDS Output Bit 8	Enable RTM LVDS Output Bit D 8
25	Enable RTM LVDS Output Bit 7	Enable RTM LVDS Output Bit D 7
24	Enable RTM LVDS Output Bit 6	Enable RTM LVDS Output Bit D 6
23	-	0
22	-	0
21	RTM LVDS Output Bit D 11 (*)	RTM LVDS Output Bit D 11 (*)
20	RTM LVDS Output Bit D 10 (*)	RTM LVDS Output Bit D 10 (*)
19	RTM LVDS Output Bit D 9 (*,#)	RTM LVDS Output Bit D 9 (*,#)
18	RTM LVDS Output Bit D 8	RTM LVDS Output Bit D 8
17	RTM LVDS Output Bit D 7	RTM LVDS Output Bit D 7
16	RTM LVDS Output Bit D 6	RTM LVDS Output Bit D 6
15	-	0
14	-	0
13	-	0
12	-	RTM Z3 TCLK select
11	-	RTM Z3 TCLK enable
10	RTM Z3 ILOCK enable	RTM Z3 ILOCK enable
9	RTM Z3 ILOCK1	RTM Z3 ILOCK1
8	RTM Z3 ILOCK0	RTM Z3 ILOCK0
7	-	0
6	-	0
5	-	RTM LVDS Input Bit D5
4	-	RTM LVDS Input Bit D4 (○)
3	-	RTM LVDS Input Bit D3 (~)
2	-	RTM LVDS Input Bit D2 (*)
1	-	RTM LVDS Input Bit D1 (*)
0	-	RTM LVDS Input Bit D0 (*)

(*): not available if board is configured for Zone 3 Class A1.1CO compatibility
(#): not available if board is configured for Zone 3 Class A1.0C compatibility
(~): not available if FW versions 8303**2**xxx, signal lines are used for µRTM-USER-I²C support, bit always returned as 0x0
(○): in FW versions 8303**2**xxx, Bit D4 used for readback of DWC8VM1 (if it's FP Interlock signal to Zone 3 supported) frontpanel interlock signal (0/1 = DWC8VM1 FP-Interlock enabled/disabled)

Note:
AMC_TCLK signal on Z3 connector is available in Z3 class A1.0C and A1.1CO only. Please refer to chapter 2.8.5 and 10.3.2.
Output signals of Interlock logic (Out0, Out1) on Z3 connector are available in Z3 class A1.1CO only. Please refer to chapter 10.3.2 and 10.4.1.

Zone 3 TCLK select table

RTM Z3 TCLK select bit	AMC_TCLK signal on Zone 3 connector derived from
0	TCLKA
1	TCLKB

Zone 3 class table

Z3 Class value	Zone 3 Class compatibility
0	A1.0
1	A1.0C
2	A1.1CO
3	reserved

7.5.33 Read DMA System Destination address (lower 32bits)

```
#define DMA_READ_DST_ADR_LO32                    0x200
```

This register holds the lower 32bits of the destination address (byte address !) in system memory into which the card will transfer data.

Bit	31-0
Function	System memory address (lower 32bits)

7.5.34 Read DMA System Destination address (upper 32bits)

```
#define DMA_READ_DST_ADR_HI32                    0x201
```

This register holds the upper 32bits of the destination address (byte address !) in system memory into which the card will transfer data.

Bit	31-0
Function	System memory address (upper 32bits)

7.5.35 Read DMA Card Memory Source address

```
#define DMA_READ_SRC_ADR_LO32                    0x202
```

This register holds the 32bit source (byte) address in the card's address space which is used to select the data source which is read from.

Bit	31-0
Function	Card address space

The address layout is:

DDR4 Memory:

```
Address 0x0 - 0x7FFFFFFF: DDR4 selected Memory
```

User DMA space:

```
Address 0x80000000 - 0x9FFFFFFF: free User DMA space
Address 0xA0000000 - 0xAFFFFFFF: Test-Bram User DMA space
Address 0xC0000000 - 0xC7FFFFFF: SFP1-BRAM User DMA space
Address 0xC8000000 - 0xCFFFFFFF: SFP2-BRAM User DMA space
Address 0xD0000000 - 0xD3FFFFFF: PORT12-BRAM User DMA space
Address 0xD4000000 - 0xD7FFFFFF: PORT13-BRAM User DMA space
Address 0xD8000000 - 0xDBFFFFFF: PORT14-BRAM User DMA space
Address 0xDC000000 - 0xDFFFFFFF: PORT15-BRAM User DMA space
```

Note: The Card address must be start on a 64-Byte boundary.
0x0, 0x40, 0x80 …

7.5.36 Read DMA Transfer length

```
#define DMA_READ_LEN                                    0x203
```

This register holds the amount of data (bytes !) which is going to be transferred.

Bit	31-0
Function	DMA Transfer length

Note: The DMA Transfer length must be a multiple of 64 Bytes.
0x40, 0x80, 0xC0 ….

7.5.37 Read DMA Control

```
#define DMA_READ_CTRL                                   0x204
```

This register starts the Read DMA process and allows to poll the transfer status.

Bit	write	read
31	unused	0
...		0
1	unused	0
0	Start DMA	DMA running

7.5.38 Readout DMA Sample byte swap

```
#define DMA_READ_BYTESWAP                                0x205
```

This register allows swapping each byte in a sample for optimizing data handling on big/little endian machines.

Example for disabled swapping:

Byte address offset:	Sample value
00	Sample 0 lo byte (LSB)
01	Sample 0 hi byte (MSB)
02	Sample 1 lo byte (LSB)
03	Sample 1 hi byte (MSB)

Example for enabled swapping:

Byte address offset:	Sample value
00	Sample 0 hi byte (MSB)
01	Sample 0 lo byte (LSB)
02	Sample 1 hi byte (MSB)
03	Sample 1 lo byte (LSB)

Bit	write	read
31	unused	0
...		0
1	unused	0
0	Byteswap enable	Byteswap enable status

7.5.39 Write DMA System Source address (lower 32bits)

```
#define DMA_WRITE_DST_ADR_LO32                    0x210
```

This register holds the lower 32bits of the destination address (byte address !) in system memory from which the card will transfer data.

Bit	31-0
Function	System memory address (lower 32bits)

7.5.40 Write DMA System Source address (upper 32bits)

```
#define DMA_WRITE_DST_ADR_HI32                    0x211
```

This register holds the upper 32bits of the destination address (byte address !) in system memory from which the card will transfer data.

Bit	31-0
Function	System memory address (upper 32bits)

7.5.41 Write DMA Card Memory Destination address

```
#define DMA_WRITE_DST_ADR_LO32                    0x212
```

This register holds the 32bit destination (byte) address in the cards address space which is used to select the data source which is written to.

Bit	31-0
Function	Card address space

The address layout is:

DDR4 Memory:

```
Address 0x0 - 0x7FFFFFFF: DDR4 selected Memory
```

User DMA space:

```
Address 0x80000000 - 0x9FFFFFFF: free User DMA space
Address 0xA0000000 - 0xAFFFFFFF: Test-BRAM User DMA space
Address 0xB0000000 - 0xB000FFFF: DAC-RAM
Address 0xC0000000 - 0xC7FFFFFF: SFP1-BRAM User DMA space
Address 0xC8000000 - 0xCFFFFFFF: SFP2-BRAM User DMA space
Address 0xD0000000 - 0xD3FFFFFF: PORT12-BRAM User DMA space
Address 0xD4000000 - 0xD7FFFFFF: PORT13-BRAM User DMA space
Address 0xD8000000 - 0xDBFFFFFF: PORT14-BRAM User DMA space
Address 0xDC000000 - 0xDFFFFFFF: PORT15-BRAM User DMA space
```

Internal the DAC RAM is mapped into the RAM address space. To write into the DAC RAM, the *'DMA write enable'* bit must set (refer chapter 7.5.21).

7.5.42 Write DMA Transfer length

```
#define DMA_WRITE_LEN                                    0x213
```

This register holds the amount of data (bytes) which is going to be transferred.

Bit	31-0
Function	DMA Transfer length

Note: The DMA Transfer length must be a power of two (min. 64 Byte).
0x40, 0x80, 0x100 ….

7.5.43 Write DMA Control

```
#define DMA_WRITE_CTRL                                   0x214
```

This register starts the Write DMA process and allows to poll the transfer status.

Bit	write	read
31	unused	0
...		0
1	unused	0
0	Start DMA	DMA running

7.5.44 Write DMA maximal number of Outstanding Requests

```
#define DMA_PC2CARD_MAX_NOF_OUTSTANDING_REQUESTS         0x215
```

This register defines the maximal number of outstanding requests during a Write DMA process. The default value is 16 (0x10).
Implemented as test feature, not needed in standard operation.

Bit	function
31	unused
...	
6	unused
5	Max Nof Outstanding Req bit 5
..	unused
0	Max Nof Outstanding Req bit 0

7.5.45 DAQ Done DMA Chain Control

```
#define DAQ_DMA_CHAIN                                   0x216
```

This register allows the chaining of the DAQ Done Signal into the DMA Start Signal.

Bit	write	read
31	unused	0
...		0
1	unused	0
0	DAQ Done DMA Start Chain enable	Chain enabled

7.5.46 IRQ Enable

```
#define IRQ_ENABLE                                      0x220
```

This register enables each interrupt source for interrupt generation. The register is implemented as a J-K register.

Bit	write	read
31	Disable User IRQ	0
30	Disable DAQ Done IRQ	0
29	unused	0
...		
18	unused	0
17	Disable Write DMA Done IRQ	0
16	Disable Read DMA Done IRQ	0
15	Enable User IRQ	User IRQ enabled status
14	Enable DAQ Done IRQ	DAQ Done IRQ enabled status
13	unused	0
...		
2	unused	0
1	Enable Write DMA Done IRQ	Write DMA Done IRQ enabled status
0	Enable Read DMA Done IRQ	Read DMA Done IRQ enabled status

7.5.47 IRQ Status

```
#define IRQ_STATUS                                    0x221
```

This register lists the latched interrupt bits for which an interrupt has been generated.

Bit	write	read
31	unused	0
...		0
16	unused	0
15	unused	User IRQ happened
14	unused	DAQ Done IRQ happened
13	unused	0
...		
2	unused	0
1	unused	Write DMA Done IRQ happened
0	unused	Read DMA Done IRQ happened

7.5.48 IRQ Clear

```
#define IRQ_CLEAR                                     0x222
```

This register clears any handled interrupts and allows the logic to generate new interrupts.

Bit	write	read
31	unused	0
...		0
16	unused	0
15	User IRQ clear	0
14	DAQ Done IRQ clear	0
13	unused	0
...		
2	unused	0
1	Write DMA Done IRQ clear	0
0	Read DMA Done IRQ clear	0

7.5.49 IRQ Refresh

```
#define IRQ_REFRESH                                   0x223
```

This register refreshes the interrupt logic. This might be needed in the case an interrupt happens while the software interrupt service routine was still handling the previous interrupt.

Bit	write	read
any	Refresh IRQ logic	0

7.5.50 Memory test mode register

```
#define MEMORY_TEST_MODE_ REGISTER 0x230
```

Test functionality only, not relevant for standard use.

7.6 External register interface

The external register interface provides the user with the possibility to implement up to 3072 32bit registers on the top level of the HDL design. The registers are embedded into the devices regular register space from address 0x400 to 0xFFF.

The External register interface consists of the following signals ():

```
COMPONENT sis_pcie_intf
port (
    ...
    reg_0x400_0xFFF_adr : out std_logic_vector(11 downto 0);
    reg_0x400_0xFFF_wr_data : out std_logic_vector(31 downto 0);
    reg_0x400_0xFFF_rd_data : in std_logic_vector(31 downto 0);
    reg_0x400_0xFFF_wr_en : out std_logic;
    reg_0x400_0xFFF_rd_en : out std_logic;
```

reg_0x400_0xFFF_adr:
: 12bit wide addressbus which selects the next register to be read from or written to.

reg_0x400_0xFFF_wr_data:
: 32bit wide databus which holds the data to be written to the addressed register.

reg_0x400_0xFFF_rd_data:
: 32bit wide databus to which the user logic must provide the read data from the addressed register.

reg_0x400_0xFFF_wr_en:
: Write enable pulse to indicate that a write request has been issued from the PCIe interface.

reg_0x400_0xFFF_rd_en:
: Read enable pulse to indicate to that a read request has been issued from the PCIe interface.

The interface is synchronous to the User Blockram DMA interface clock (dma_bram_clk). See 7.7.

See the following graphs on how the device expects user logic to interact with the interface.

Note: the following graphs are taken from the SIS8300L2 manual. These will be updated with the manual version!

The signals from top to bottom are:

1. register read pulse
2. register write pulse
3. register address bus (8 bits wide)
4. register data write bus (32 bits wide)
5. register data read bus (32 bits wide from user defined register logic)

A read request from the external register space causes the signals to change as follows:

The address bus changes its value to the current selected register.

The read enable signal is pulsed for 1 clockperiod.

The user data from the external register is expected to be valid 1 clockperiod after the read pulse.

The signals from top to bottom are:

1. register read pulse
2. register write pulse
3. register address bus (8 bits wide)
4. register data write bus (32 bits wide)

A write request to the external register space causes the signals to change as follows:

The address and write data bus change their values to the current selected register and the new register data.

The write enable signal is pulsed for 1 clockperiod.

7.7 User Blockram DMA Interface

The User Blockram DMA interface consists of the following signals:

```
COMPONENT sis_pcie_intf
port (
      ..
      dma_bram_clk : out std_logic; -- pcie user side clock
      dma_bram_read_add_latency : in std_logic_vector(1 downto 0);
      dma_bram_read_en : out std_logic;
      dma_bram_read_addr : out  std_logic_vector(31 downto 0);
      dma_bram_read_data : in  std_logic_vector(255 downto 0);

      dma_bram_write_en : out std_logic;
      dma_bram_write_addr : out  std_logic_vector(31 downto 0);
      dma_bram_write_data : out  std_logic_vector(255 downto 0);
```

dma_bram_clk:
Free running 125MHz clock to which the data and control signals are synchronous to

dma_bram_read_addr:
32bit wide addressbus which is mapped over the 512MB of onboard sample RAM. The addresses ranges from 0x00000000 to 0x1FFFFFFF (512MB)

dma_bram_read_en:
(Optional) Read enable pulse for connected logic. The read enable pulse is valid 1 clockperiod before the data is expected to be valid on the databus.

dma_bram_read_data:
256bit wide (due to PCIe endpoint design) databus which holds the data to be transmitted over PCIe. The data needs to be valid 1 clockperiod after dma_bram_read_en is valid.

Note:
This interface was built to be directly able to connect to a Xilinx CoreGen generated blockram module with a 256bit wide read bus. Due to PCIe endpoint design constraints the user has to read the blockram via DMA in 32byte steps.

Graphs (screenshots of the timings) and the description of the “dma_bram_write” bus will come with the next update of the manual.

7.8 User Interrupt Interface

The User Interrupt interface consists of the following signals.

```
COMPONENT sis_pcie_intf
port (
      ..
      user_irq : in std_logic;
      user_irq_clear : out std_logic;
```

user_irq:

User interrupt pulse input to the PCIe endpoint. The connected logic has to supply a pulse of at least 1 clockperiod in length to trigger the interrupt logic. If the driver/user software has enabled the corresponding interrupt line, an interrupt is generated on the PCIe interface.

user_irq_clear:

A 1 clock period wide pulse which is triggered, when the driver software has serviced the issued interrupt in the interrupt service routine. This pulse may be used to reset any logic that depends on user feedback through the software interface.

The interface is synchronous to the User Blockram DMA interface clock. See section 7.7.

8 µRTM management

Connected µRTMs shall be compliant to the PICMG MTCA.4 specification in a way that they must have an on board I²C EEPROM (on address 0x50) and a NXP PCF8574-compatible port expander (on address 0x7C).

Required port expander connection map for normal operation:

Port pin	Function
P0	Hot Swap Handle (low active)
P1	Hot Swap LED (Blue, low active)
P2	LED 1 (Red, low active)
P3	LED 2 (Green, low active)
P4	PowerGood (low active)
P5	Reset (low active)
P6	PowerEnable (low active)
P7	EEPROM Write Protect

The EEPROM shall contain any relevant device information (FRU records) about the µRTM (refer to PICMG AMC.0). Additionally the EEPROM shall contain the new record types defined in PICMG MTCA.4.

In order to be able to decide whether a connected µRTM is compatible to the SIS8300-KU the µRTM record shall contain one of the Zone 3 Identifier listed in the table below.

Supported Zone 3 Identifier (Interface Identifier OEM):

IANA PEN (Private enterprise number)	Interface Identifier OEM	Description
0x0092BD (37565)	0x83000003	SIS8300-KU Zone 3 - compatibility
0x00053F (1343)	0x08020000	DESY DWC Zone 3 compatibility
0x0092BD (37565)	0x89000001	SIS8900 Zone3 - v1 compatibility

9 FPGA Firmware upgrade

FPGA firmware update can be done in three different ways.

1. JTAG connection
2. PCI Express interface access to Xilinx configuration prom
3. MMC access to Xilinx configuration prom via IPMI commands

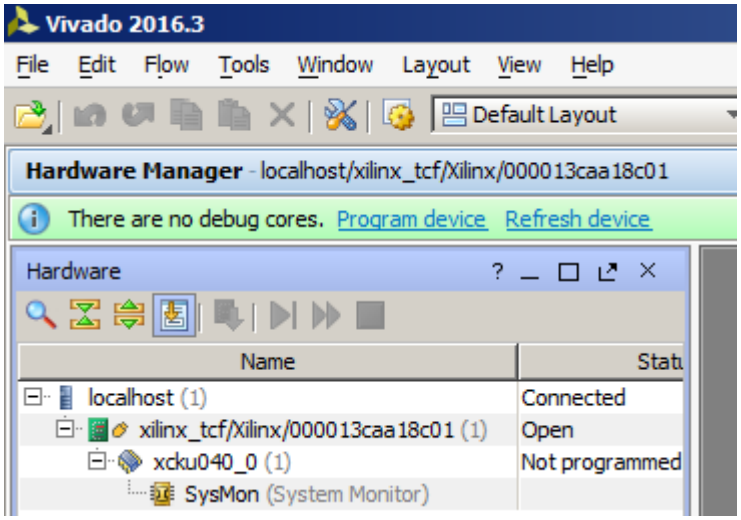
First is the only way of all needing an additional hardware tool like XILINX Platform cable USB device. Also Xilinx Vivado software must be installed on USB connected computer. On SIS8300-KU with intact Xilinx firmware only, the second way is the fastest of all. Second and third way uses different LINUX tools (which can be found on the product DVD) .

9.1 JTAG Firmware Upgrade procedure

1. Connect PC and SIS8300-KU CON100 with "XILINX Platform cable USB" device
2. Set Jumper J604 to open position
2. Power up unit (Hot Swap Handle pushed)

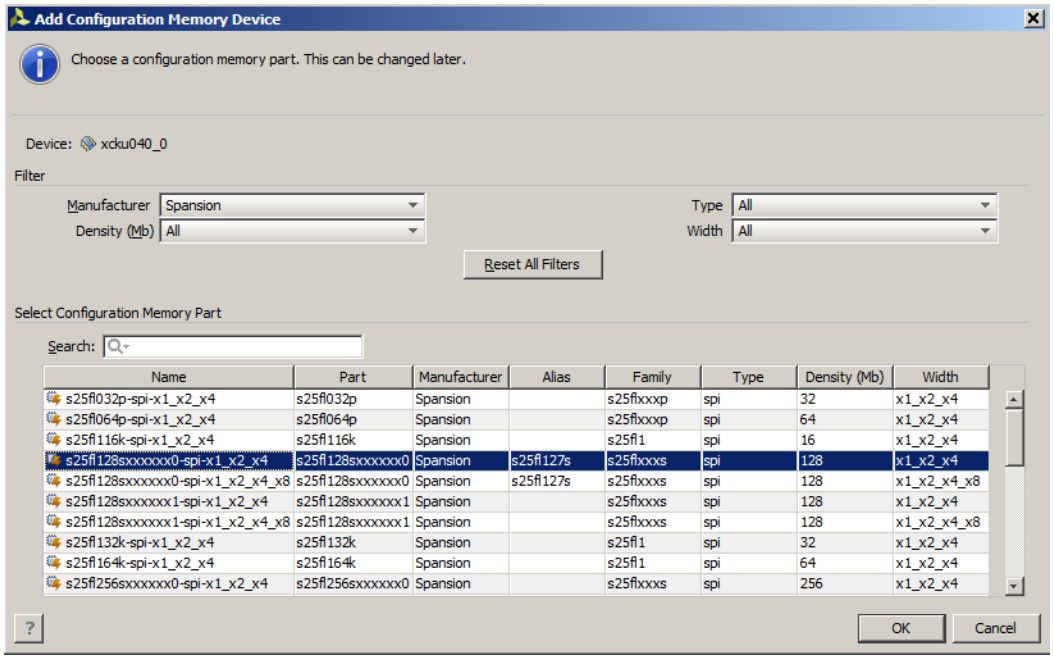
9.1.1 Use of Xilinx Vivado Hardware Manager

Start Vivado and open Hardware Manager. To connect hardware with manager, click with mouse on "Open target" string located in single row above empty main window and select "Auto connect". Now in left side hardware window "xcku040_0" device must be recognised how to see in screenshot below.

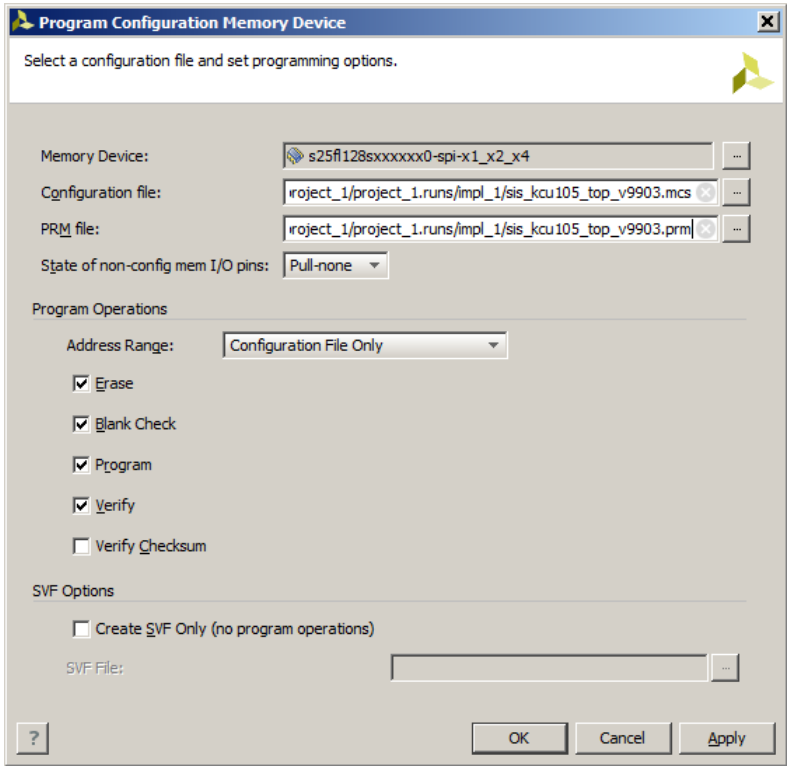


Right click with mouse on xcku040_0 row and select "Add Configuration Memory Device". In next step limit the selectable memory devices to "Manufacturer" Spansion.

Now please select configuration memory and confirm with "OK" like to see in screenshot below.



Now in left side hardware window selected configuration memory part appears.
Right click with mouse on s25fl128s... device and select "Program Configuration Memory Device...". In new dialog window please select configuration file and PRM file from your file residing location. It's necessary to set check buttons "Erase" an "Program" also at minimum.
A typical dialog setting is to see in screenshot below.



Confirming the dialog with "OK" starts selected program operations.
Depending on selected operations whole process can take several minutes.

Note: The Basic and Second configuration memory (FLASH SPI EEPROM) is selected via ipmitool as desribed in section 2.4.

9.2 Linux tool 'flashupdater'

This method can not be used on cards with flawed firmware.
To be able to use this tool, it is required that the driver is loaded and the devices are mounted properly. Use the commands below for verification:

`# lsmod | grep sis`	Result (if driver loaded): `sis8300drv`
`# ls /dev/ | grep sis`	Result: for each installed device one entry (`sis8300-0` e.g.)

Now the update can be started:

```
./flashupdate <path/to/file>.bin
```

9.3 Linux tool 'ipmitool'

The last from four methods of upgrading is to upload FPGA firmware via IPMI.
This can be made with LINUX software ipmitool.

First it is required to convert the FPGA firmware bit file into an IPMI compatible hpm file.
For this the Windows command line tool bin2hpm.exe can be used:

```
> bin2hpm /bit /compress <file_name>.bit
```

The tool generates the hpm file <file_name>.rle.hpm, which can be transferred now to the SIS8300L2 by using the ipmitool:

Syntax:
```
ipmitool -H <IP_or_name_of_MCH> -P "" -B 0 -b 7 -T 0x82 -t <slot_number>
hpm upgrade <file_name>.rle.hpm force
```
slot_number: 0x72 (1 slot), 0x74 (2 slot), 0x76 (3 slot)

for example:
```
ipmitool -H 192.168.115.62 -P "" -B 0 -b 7 -T 0x82 -t 0x74 hpm upgrade
sis8300L2_200E.rle.hpm force
```

9.4 Linux workaround for PCIe-Hot-Plug

During the firmware development phase it may be of interest to re-establish PCIe connectivity to the SIS8300-L without a power cycle of the crate. A workaround is given in the command sequence below.

1. Open a terminal and deactivate the Struck device driver:

   ```
   sudo rmmod sis8300drv
   ```

2. Change into the root space:

   ```
   sudo su
   ```

3. Disable the PCIe link of the target device slot (e.g. slot number 5):

   ```
   echo 0 > /sys/bus/pci/slots/5/power
   ```

   The link LED (L1) goes off.

4. 5. Reactivate the PCIe slot:

   ```
   echo 1 > /sys/bus/pci/slots/5/power
   ```

   The link LED (L1) goes back on.

10 Appendix

10.1 Power Consumption

The module current requirements record is defined to 5.0A and includes the current budget for an attached µRTM also. The currents drawn by the SIS8300-KU are listed in the table below.

Voltage	Current
3.3 V	70 mA
12 V	3.5 A

Current on 12V is a typical value during normal operation. It can vary depending on the loaded firmware design.

10.2 Ordering options

The available part numbers are listed in the table below.

Struck part number	Part name and configuration
05756	**SIS8300-KU 8AC2DC DZ3** Channel 0-7 AC ADC Input Coupling, 62MHz BW, Channel 8-9 DC ADC Input Coupling, 140MHz BW, AD8139, DAC to Z3, with FP SMA and RJ45, Zone 3 Class A1.1CO

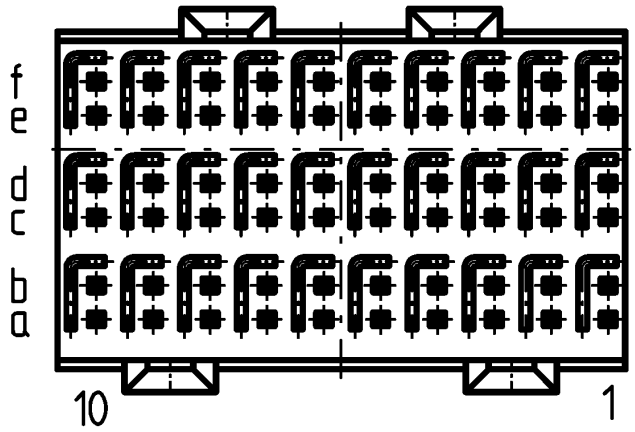
10.3 SFP+ Accessories

Find below a table of components that were used with the card during the development phase and/or are used in testing the hardware.

Struck part number	Manufacturer part number	Description
05770	PLRXPL-SC-S43-22-N Lumentum	10 GBit/s LC Link Medium
05768	33070740050 Eurolan	LC/LC Duplex 50/125 µm Fiber 5m OM3

10.4 Zone 3 connectors J30 and J31

J30 and J31 are 90 pin right angle female connectors providing 30 contact pairs each (60 signal contacts and 30 ground contacts). Every contact pair is surrounded by a “L” shaped male shield blade. The shielding contact is designated with the names of the corresponding signal pair (signal pin a and b is affiliated with shielding contact ab e.g.). The picture below shows the connector contact layout as seen from the rear side of the board.



10.4.1 J31 connector pin assignments

The J31 connector routes the differential analog input signals of the ADC channels and ground to the µRTM. The characters “TF” in signal names stand for signals to the AC coupled transformer input stages. In same fashion “PA” stands for DC coupled preamplifier input stage.

Col → Row ↓	ef	f	e	cd	d	c	ab	b	a
10	GND	CH0_PA-	CH0_PA+	GND	GND	GND	GND	CH0_TF-	CH0_TF+
9	GND	CH1_TF-	CH1_TF+	GND	GND	GND	GND	CH1_PA-	CH1_PA+
8	GND	CH2_PA-	CH2_PA+	GND	GND	GND	GND	CH2_TF-	CH2_TF+
7	GND	CH3_TF-	CH3_TF+	GND	GND	GND	GND	CH3_PA-	CH3_PA+
6	GND	CH4_PA-	CH4_PA+	GND	GND	GND	GND	CH4_TF-	CH4_TF+
5	GND	CH5_TF-	CH5_TF+	GND	GND	GND	GND	CH5_PA-	CH5_PA+
4	GND	CH6_PA-	CH6_PA+	GND	GND	GND	GND	CH6_TF-	CH6_TF+
3	GND	CH7_TF-	CH7_TF+	GND	DAC1-	DAC1+	GND	CH7_PA-	CH7_PA+
2	GND	CH8_PA-	CH8_PA+	GND	GND	GND	GND	CH8_TF-	CH8_TF+
1	GND	CH9_TF-	CH9_TF+	GND	DAC2-	DAC2+	GND	CH9_PA-	CH9_PA+

10.4.2 J30 connector pin assignments

The J30 connector is used to route power, data and system management pins to the µRTM board. Pin Assignment is conform to DESY's Zone 3 Connector Pin Assignment Recommendation Class A1 and will be manufacturer configured as Subclass A1.0, A1.0C or A1.1CO compliant.

Pin Assignment for Class A1.0

Col → Row ↓	ef	f	e	Cd	D	c	ab	b	a
10	GND	GND	GND	GND	GND	GND	GND	GND	GND
9	GND	CLK1-	CLK1+	GND	nc	nc	GND	CLK0-	CLK0+
8	GND	CLK5-	CLK5+	GND	CLK2-	CLK2+	GND	CLK4-	CLK4+
7	GND	GND	GND	GND	GND	GND	GND	GND	GND
6	GND	D11-	D11+	GND	D10-	D10+	GND	D9-	D9+
5	GND	D8-	D8+	GND	D7-	D7+	GND	D6-	D6+
4	GND	D5-	D5+	GND	D4-	D4+	GND	D3-	D3+
3	GND	D2-	D2+	GND	D1-	D1+	GND	D0-	D0+
2	GND	TMS	TDI	GND	SCL	MP+3.3V	GND	PWR+12V	PWR+12V
1	GND	TDO	TCK	GND	SDA	PS#	GND	PWR+12V	PWR+12V

Pin Assignment for Class A1.0C

Col → Row ↓	ef	f	e	Cd	D	c	ab	b	a
10	GND	GND	GND	GND	GND	GND	GND	GND	GND
9	GND	CLK1-	CLK1+	GND	nc	nc	GND	CLK0-	CLK0+
8	GND	CLK5-	CLK5+	GND	CLK2-	CLK2+	GND	CLK4-	CLK4+
7	GND	GND	GND	GND	GND	GND	GND	GND	GND
6	GND	D11-	D11+	GND	D10-	D10+	GND	AMC_TCLK-	AMC_TCLK+
5	GND	D8-	D8+	GND	D7-	D7+	GND	D6-	D6+
4	GND	D5-	D5+	GND	D4-	D4+	GND	D3-	D3+
3	GND	D2-	D2+	GND	D1-	D1+	GND	D0-	D0+
2	GND	TMS	TDI	GND	SCL	MP+3.3V	GND	PWR+12V	PWR+12V
1	GND	TDO	TCK	GND	SDA	PS#	GND	PWR+12V	PWR+12V

Pin Assignment for Class A1.1CO

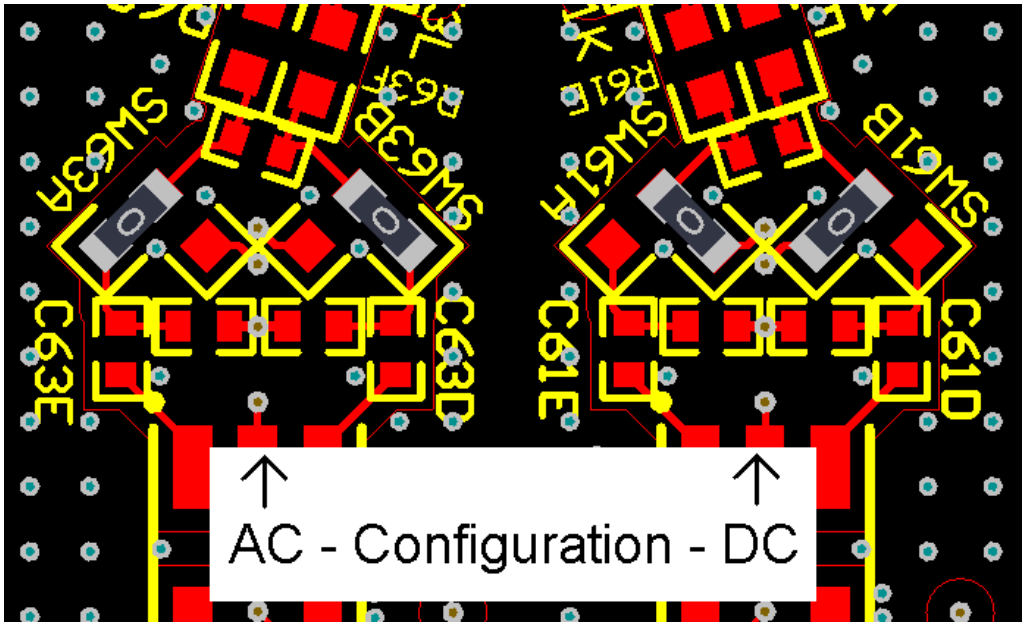
Col → Row ↓	ef	f	e	Cd	d	c	ab	b	a
10	GND	GND	GND	GND	GND	GND	GND	GND	GND
9	GND	CLK1-	CLK1+	GND	nc	nc	GND	CLK0-	CLK0+
8	GND	CLK5-	CLK5+	GND	CLK2-	CLK2+	GND	CLK4-	CLK4+
7	GND	GND	GND	GND	GND	GND	GND	GND	GND
6	GND	OUT1-	OUT1+	GND	OUT0-	OUT0+	GND	AMC_TCLK-	AMC_TCLK+
5	GND	D8-	D8+	GND	D7-	D7+	GND	D6-	D6+
4	GND	D5-	D5+	GND	D4-	D4+	GND	D3-	D3+
3	GND	SFP-TX-	SFP-TX+	GND	SFP-RX-	SFP-RX+	GND	SFP-CLK-	SFP-CLK+
2	GND	TMS	TDI	GND	SCL	MP+3.3V	GND	PWR+12V	PWR+12V
1	GND	TDO	TCK	GND	SDA	PS#	GND	PWR+12V	PWR+12V

Note: SFP signal pins of row 3 are not connected (open)
Different signals of class A1.0C and A1.1CO to class A1.0 are shown in colour

For more information about Zone 3 recommendation, please refer to
http://mtca.desy.de/resources/zone_3_recommendation/index_eng.html

10.5 Note on AC/DC input stage selection

The AC (transformer) or DC (operational amplifier) input path is selected on the SIS8300-KU card via 0603 solder bridges as illustrated for channels 0 and 1 on the screenshot below.



The designators for all channels can be found in the table below.

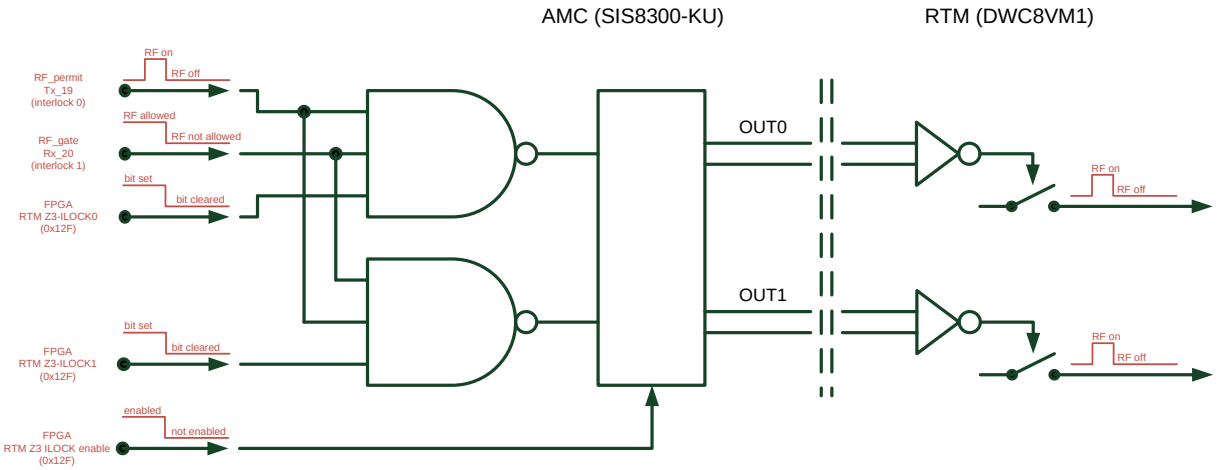
ADC Channel	Designator	Input for ADC
0	SW63A	ADC_CH0+
0	SW63B	ADC_CH0-
1	SW61A	ADC_CH1+
1	SW61B	ADC_CH1-
2	SW58A	ADC_CH2+
2	SW58B	ADC_CH2-
3	SW56A	ADC_CH3+
3	SW56B	ADC_CH3-
4	SW53A	ADC_CH4+
4	SW53B	ADC_CH4-
5	SW51A	ADC_CH5+
5	SW51B	ADC_CH5-
6	SW48A	ADC_CH6+
6	SW48B	ADC_CH6-
7	SW46A	ADC_CH7+
7	SW46B	ADC_CH7-
8	SW43A	ADC_CH8+
8	SW43B	ADC_CH8-
9	SW41A	ADC_CH9+
9	SW41B	ADC_CH9-

Note: Keep in mind that filtering components (like antialiasing) may be involved also for your particular application.

10.5.1 Interlock

An interlock mechanism is implemented on the SIS8300-KU for operation with Vectormodulator cards like the DWC8VM1 and DS8VM1. The OUT0 and OUT1 Zone 3 signals are generated from the logical combination of the backplane signals TX_19 (RF permit), RX_20 (RF gate) and the FPGA ILOCK signals.

The interlock scheme is shown in the diagram (courtesy of DESY) below.

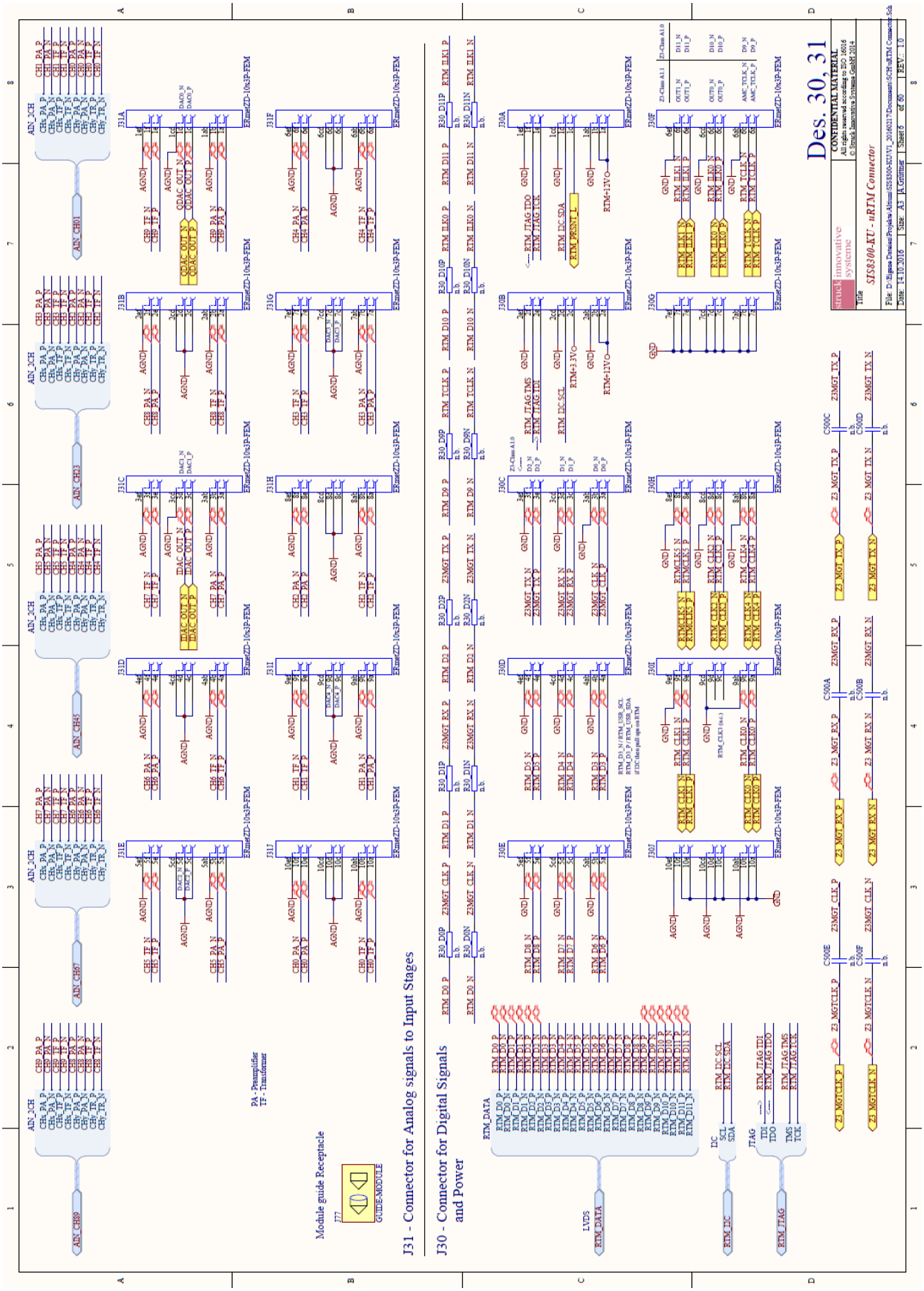


Interlock signals are available for SIS8300-KU with Zone 3 class compatibility A1.1CO only. Please refer to section 10.4.2.

Please refer to application note "Interlock use with SIS8300-L and SIS8300-L2" for additional information. It can be found on the Struck product DVD under: sisdvd_xxxxxx\sis8xxx and DWC\sis8300L2\doc\SIS8300-L_L2-AT-311214-1-V110-Interlock.pdf

SAFETY NOTE: make sure, that you have the proper interlock scheme for your application. Struck Innovative Systeme GmbH does not assume any liability for improper configuration.

10.6 Zone 3 connector schematic



10.7 MMC Readme file

The license to use the DESY MMC code is granted to Struck Innovative Systeme under license contract LV92. Struck and its' end users are bound to adhere to the contents and conditions of the following Readme file (which can be found on our product DVD also):

```
DESY MMC end user licence notice v1.0

1

1. The MMC firmware on this device is based on the DESY MMC firmware package.
It has been adapted to the device by the device vendor.

2. DESY MMC firmware package is written for operation at DESY and is provided as-is to the device vendor,
without any warranty including all implied warranties of merchantability and fitness.
In no event shall DESY be liable for any special, direct indirect, or consequential damages,
or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract,
negligence or other tortious action, arising out of or in connection with the use or performance of this
software.

3. Claims for damages of any nature against DESY are expressly excluded,
except in case of gross negligence or wilful misconduct.
DESY shall not be liable for any indirect damages or damages caused by assistants.

4. The firmware is distributed as binary code, either on the MMC or as a data package for a firmware
update.
The end user may not disassemble or reverse-engineer any parts of the MMC firmware.

5. The end user may keep backup copies of the MMC firmware,
or copies which are necessary to install the MMC firmware in the end user's MTCA infrastructure.
The end user may not distribute the MMC firmware to a third party.

Warning: The MMC firmware has been adapted to a specific device.
DO NOT USE MMC FIRMWARE ON A DEVICE IT HAS NOT BEEN ADAPTED TO.
USING THE WRONG FIRMWARE CAN CAUSE DAMAGE TO THE DEVICE OR THE MTCA INFRASTRUCTURE IT IS CONNECTED TO.
When in doubt contact the device vendor.
```

11 Index